



Universidad
Carlos III de Madrid

Departamento de Informática
Ingeniería en Informática

PROYECTO DE FIN DE CARRERA

Sistema de geolocalización de personas en interiores de
grandes edificios (Plataforma IOS)



Autor: Óscar Martín Samarín

Tutor: Dr. Javier García Guzmán

Leganés,

Agradecimientos

El desarrollo de la aplicación documentada en esta memoria me ha permitido aprender mucho sobre las tecnologías que se usan y que me gustaría seguir usando en mi carrera profesional, además representan el final de un ciclo de aprendizaje tutorado que en mi caso ha sido muy largo, habiendo pasado por la Universidad de Las Palmas de Gran Canaria, la Universidad de La Laguna y finalmente La Universidad Carlos III de Madrid. Un periodo universitario que empezó en 2001 en el que he aprendido de la experiencia y recibido el soporte de muchas personas a las que me gustaría hacer llegar mi gratitud.

Soy de Santa Cruz de Tenerife y en 2001 comencé Ingeniería informática de sistemas en la Universidad de Las Palmas de Gran Canaria. Tenía que dejar mi casa y mudarme a otra isla si quería estudiar allí. Esta universidad puede estar muy orgullosa de los que ahí trabajan, de sus planes de estudio y de su manera de enseñar. Su biblioteca tiene buenos ejemplares como la UC3M. En ella casi aprendo a ser un hacker y conocí la trinchera de la informática. Hice grandes amigos y hoy por hoy muchos son los mejores programadores que conozco. Tengo que decir que el nivel técnico que había en esa escuela no he vuelto a conocerlo en ninguna otra, aunque estoy seguro que fue debido a que no coincidí con los más hambrientos en el resto de universidades. Ese hambre por aprender informática y lo aprendido me ayudó a descubrir la necesidad de estudiar el ciclo superior que termino con este trabajo y por ello quiero agradecer a mis compañeros de la ULPGC y la Escuela Superior de Informática que dejaron que aprendiera de ellos.

Al acabar la carrera volví a Tenerife, donde encontré un trabajo distinto a las becas que empezaba a enseñarme lo que era el negocio del software ya que hasta entonces todo parecían prácticas en un “entorno laboral real”. Trabajé tanto en consultoras como en clientes de consultoras pero quería saber más. Empezaba a desarrollar aplicaciones complejas en las que participaban varios desarrolladores que teníamos que cumplir con un contrato. Pensé que esa organización, esa gestión y ese diseño tenían una razón de ser y pensé que eran mejorables. Se me ocurrió que quizá las asignaturas de ingeniería del software que se veían en el ciclo superior me ayudarían a aprender a diseñar mejor software y entender mejor el negocio de la construcción del software por lo que me matriculé en la Universidad de La Laguna. Me matriculé sólo de Ingeniería del Software I, Redes I e Inteligencia artificial. Me llevé un desengaño con Ingeniería del software porque no me enseñaban a diseñar y construir software sino a documentarlo, lo cual no dejaba de tener su importancia. Inteligencia artificial me pareció un descubrimiento muy interesante. Esta universidad no tenía más asignaturas de inteligencia artificial por lo que el plan de estudios se me hizo poco atractivo y me vi empujado a buscar otras universidades donde enseñasen inteligencia artificial. Es entonces cuando encontré la Universidad Carlos III de Madrid.

En teoría la UC3M era la que cumplía todos los requisitos que necesitaba. Y en la práctica no me decepcionó. Traté de cursar las especialidades de ingeniería del software y la de inteligencia artificial pero finalmente completé los créditos de inteligencia artificial. Sus asignaturas me resultaban más motivantes y los profesores parecían conocer secretos que ningún otro informático sabía. Será difícil que durante la carrera laboral llegue a poner en práctica algo que aprendiese en la especialidad por lo que tengo que agradecer a los

profesores que tutorizasen unas prácticas que de otro modo nunca hubiera hecho y nunca hubiera descubierto.

Traté de hacer un proyecto de fin de carrera que tuviera que ver con la inteligencia artificial. Comencé a conocer al robot Nao y el Sistema operativo ROS pero la disponibilidad del laboratorio y de las personas que podían ayudarme era escasa y pensé que me lo estaban poniendo más difícil de la cuenta, había vuelto a trabajar y mi disponibilidad de tiempo tampoco era la que mejor se adecuaba a tal fin. Me enteré entonces que en el SEL se estaban tutorizando proyectos de fin de carrera para desarrollar aplicaciones para iOS, por lo que fui a hablar con el profesor D. Javier García Guzmán para contarle que quería hacer una aplicación para el iPhone. Esta vez tenía mi iPhone y mi MacBook por lo que entendía que tendría menos impedimentos para llevarlo a cabo. Después de contarme sus necesidades y que me decidiera por una, el tutor me dio la libertad para hacerlo de la manera que quise sin poner ninguna traba. Fue un cambio radical entre el primer proyecto que empecé y éste por lo que le estoy tremendamente agradecido a Javier por ser mi tutor, estoy seguro que le hubiera gustado que lo terminase antes, pero tampoco ha mostrado queja alguna y me ha dejado trabajar a mi ritmo.

Hice buenos amigos en la UC3M. Ellos me ayudaron con las prácticas y con sus distintos puntos de vista para enseñarme otra manera de hacer y de entender las cosas. Disfruté mucho cambiando impresiones con ellos porque aprendí que siendo buenas todas las opciones hay que elegir siempre y no siempre hay que elegir lo mismo. Y es que la solución buena nunca era la perfecta y que todos entendemos la perfección de manera distinta. A todos mis amigos, gracias.

Los penúltimos agradecimientos se los voy a dedicar a mis compañeros de trabajo en Airbus. No sólo me han dado feedback sobre la aplicación sino que me apoyaron y cubrieron mi ausencia para que pidiese una excedencia de mes y medio para que terminase de una vez este proyecto de fin de carrera. Airbus es una empresa como no había conocido con anterioridad, contempla en su normativa interna que los trabajadores podamos disfrutar estos permisos para finalizar estudios si nuestro jefe nos autoriza. Gracias, por tanto, a Angel Mangas por ficharme hace tres años y autorizar mi excedencia.

Y muchísimas gracias a mi familia. Tengo que darles las gracias porque sin su apoyo incondicional y sus buenos consejos nunca hubiera terminado la carrera. Me han dado todo lo que han podido para que pudiera estudiar lo que quise y donde quise sin cuestionar mis deseos y apetencias. Ellos han hecho lo posible por participar de forma activa en mi formación, incluso en este proyecto de fin de carrera. Mi hermana, Irene, se ha encargado con mucho gusto de hacer el logotipo de la aplicación. Creo que se esforzó más de lo que le hubiera pedido. Con todo lo que he contado aquí, estoy seguro que es fácil entender que unas palabras de agradecimiento no bastan para expresar la gratitud que siento hacia ellos.

Contenido

1	INTRODUCCIÓN.....	10
1.1	MOTIVACIÓN DEL PROYECTO.....	10
1.1.1	Escenario 1: Disponibilidad y presencia del profesor.....	10
1.1.2	Escenario 2: Presencia alumnos.....	11
1.1.3	Escenarios generales.....	11
1.2	OBJETIVO.....	12
1.3	METODOLOGÍA.....	12
1.4	CONTENIDO DEL DOCUMENTO.....	13
1.5	REFERENCIAS.....	14
2	ANÁLISIS DEL ESTADO DE LA CUESTIÓN.....	18
2.1	APLICACIONES SIMILARES.....	18
2.1.1	Whatsapp.....	18
2.1.2	Buscar a mis amigos.....	19
2.1.3	Google Maps.....	20
2.1.4	UC3M Mapas.....	21
2.1.5	Waze.....	22
2.1.6	Google+.....	22
2.1.7	Facebook.....	23
2.1.8	Foursquare.....	24
2.1.9	Swarm.....	25
2.1.10	Comparativas y conclusiones.....	26
2.2	ANÁLISIS DE TECNOLOGÍAS DISPONIBLES.....	28
2.2.1	Heroku.....	28
2.2.2	Amazon AWS.....	28
2.2.3	Google cloud.....	29
2.2.4	Comparativas y conclusiones.....	29
3	ESPECIFICACIÓN DE REQUISITOS SOFTWARE.....	32
3.1	INTRODUCCIÓN.....	32
3.1.1	Propósito.....	32
3.1.2	Ámbito.....	32
3.1.3	Definiciones, Acrónimos y Abreviaturas.....	32
3.1.4	Referencias.....	33
3.1.5	Visión general del documento.....	33
3.2	DESCRIPCIÓN GENERAL.....	33
3.2.1	Perspectiva del producto.....	33
3.2.2	Funciones del producto.....	35
3.2.3	Características de los usuarios.....	36
3.2.4	Restricciones.....	36
3.2.5	Suposiciones y dependencias.....	36
3.2.6	Requisitos futuros.....	36
3.3	REQUISITOS ESPECÍFICOS.....	37
3.3.1	Interfaces externas.....	37
3.3.2	Interfaces de usuario.....	37
3.3.3	Funciones.....	39

3.3.4	<i>Requisitos de rendimiento</i>	46
3.3.5	<i>Restricciones de diseño</i>	47
3.3.6	<i>Atributos del sistema</i>	47
4	DISEÑO	50
4.1	ALTERNATIVAS DE DISEÑO.....	50
4.2	DISEÑO.....	51
4.2.1	<i>Find My Mates Web Service</i>	51
4.2.2	<i>Find My Mates App</i>	59
5	PRUEBAS DE SISTEMA	95
5.1	US-01: VER CONTACTOS.....	95
5.2	US-02: AÑADIR AMIGO.....	95
5.3	US-03: ELIMINAR CONTACTO.....	97
5.4	US-04: VER EDIFICIOS DE AUTOCHECKIN.....	98
5.5	US-05 AÑADIR EDIFICIOS.....	98
5.6	US-06: ELIMINAR EDIFICIO.....	99
5.7	US-07: CHECKIN MANUAL.....	99
5.8	US-08: AUTOCHECKIN.....	100
5.9	US-09 DESACTIVAR AUTOCHECKIN.....	101
5.10	US-10: ACTIVAR AUTOCHECKIN.....	101
5.11	US-11: HACER LOGIN.....	101
5.12	US-12: HACER LOGOUT.....	102
5.13	US-13 VER CHECKINS.....	103
5.14	US-14: VER CHECKINS EN UN GIS.....	104
5.15	MATRIZ DE TRAZABILIDAD.....	104
6	CONCLUSIONES	108
6.1	LÍNEAS FUTURAS.....	109
6.1.1	<i>Repositorios de información geográfica</i>	109
6.1.2	<i>Redes sociales</i>	109
6.1.3	<i>Privacidad</i>	109
6.1.4	<i>Idiomas</i>	109
6.1.5	<i>Integración</i>	109
6.1.6	<i>QR checkin</i>	109
6.1.7	<i>Localizaciones en interiores</i>	109
6.1.8	<i>Check-outs</i>	109
6.1.9	<i>Notificaciones</i>	109
	ANEXO I: MANUAL DE USUARIO	112
	REQUISITOS DE FUNCIONAMIENTO.....	112
	PRIMEROS PASOS.....	113
	<i>Permisos de localización</i>	113
	<i>Loggin</i>	114
	AJUSTES.....	115
	CONTACTOS.....	116
	<i>Añadir contactos</i>	116
	<i>Borrar contactos</i>	117

<i>Mapa</i>	117
EDIFICIOS	119
<i>Añadir edificios</i>	119
<i>Eliminar edificios</i>	120
CHECKIN	120
ANEXO II: PLAN DE TRABAJO	124
PLANIFICACIÓN	124
RECURSOS EMPLEADOS	125
<i>Recursos hardware</i>	125
<i>Recursos software</i>	125
ANEXO III: STORY BOARD	129
CONTACTS	129
BUILDINGS	130
CHECKIN	131
SETTINGS	132

Capítulo 1: Introducción

Contenido

1	INTRODUCCIÓN.....	10
1.1	MOTIVACIÓN DEL PROYECTO.....	10
1.1.1	<i>Escenario 1: Disponibilidad y presencia del profesor.....</i>	<i>10</i>
1.1.2	<i>Escenario 2: Presencia alumnos.....</i>	<i>11</i>
1.1.3	<i>Escenarios generales.....</i>	<i>11</i>
1.2	OBJETIVO.....	12
1.3	METODOLOGÍA.....	12
1.4	CONTENIDO DEL DOCUMENTO.....	13
1.5	REFERENCIAS.....	14

1 Introducción.

En este capítulo se explicarán cuáles han sido las motivaciones fundamentales para desarrollar el presente proyecto, así como los objetivos que pretende alcanzar. En último lugar se da una breve descripción del contenido de este documento.

1.1 Motivación del proyecto.

El proyecto viene motivado por el interés del desarrollador en aprender sobre tecnologías móviles. Los smartphones tienen unas características especiales, como son la incorporación de un receptor de GPS, sensores de movimiento y conectividad permanente a internet en cualquier lugar, que les permiten resolver problemas que otros dispositivos no pueden. A su vez estas características especiales añaden nuevos problemas de ingeniería como la gestión de la batería o tareas en background que merecen atención para el desarrollo de este proyecto.

El problema que detectado para resolver con un smartphone es la gestión de la presencia de las personas dentro de la UC3M dado que no siempre es fácil conocer con antelación suficiente donde se va a estar para poder coincidir con otros miembros de la UC3M.

Los sistemas de gestión de presencia actuales consisten básicamente en una fichada manual en donde la presencia de la persona sólo la conoce quien tenga acceso al sistema en el que se centralizan los datos, y no siempre se pueden verificar en tiempo real. Además la persona que hace esta fichada no puede definir quién puede ver donde se encuentra y quién no. Un ejemplo de un sistema que permite hacer esta gestión es SAP. Otras aplicaciones más sociales que permiten publicar la posición del usuario serán analizadas en el siguiente capítulo.

Este proyecto pretende desarrollar un sistema en el que la fichada o publicación de ingreso en un edificio es automática y distribuida, de manera que estará disponible para todos los clientes del sistema que autorice el autor de la fichada en tiempo real. En términos prácticos en la UC3M se contemplan varios escenarios.

1.1.1 Escenario 1: Disponibilidad y presencia del profesor.

En el día a día de la vida universitaria se acude a las aulas o laboratorios donde se imparten clases. Estas clases no son siempre las mismas para las mismas asignaturas, hay distintos grupos y grupos de prácticas. Además existen los cambios puntuales de aula. Esto causa ocasionales despistes que con el uso de esta aplicación solo causarían pequeños retrasos en la llegada a la clase al conocerse donde se encuentra el profesor que la imparte.

Los profesores tienen una serie de horas planificadas para tutorías en un despacho. En algunas ocasiones se ve conveniente tener la tutoría en otra sala o para toda una clase y como esto es una actividad no planificada a veces se ve la dificultad de conocer el lugar exacto donde llevarla a cabo. Con este sistema simplemente el usuario de esta aplicación tiene que estar a la hora acordada en el campus acordado y al abrir la aplicación verá donde se encuentra el profesor.

Otros profesores pueden preferir atender tutoría siempre que se encuentren en su despacho sin ajustarse a un horario predeterminado, dado que tiene pendiente ajustar su agenda con actividades variables. Con esta aplicación un alumno podría

saber en tiempo real si el profesor se encuentra en el despacho o no, ahorrando tiempo al alumno en sus desplazamientos.

La aplicación que se desarrolla también es especialmente útil para los visitantes de esta universidad cuando estos no conocen el campus al que acuden. Solo conoce a la persona que va a visitar. Con esta aplicación, se puede ver en donde se encuentra la persona a visitar y como llegar a ella mediante un plano.

1.1.2 Escenario 2: Presencia alumnos.

En el día a día de la vida universitaria se acude a aulas o laboratorios donde se imparten las clases. Estas clases no son siempre en las mismas aulas o mismos laboratorios para las mismas asignaturas, hay distintos grupos y grupos de prácticas. Además existen los cambios puntuales de aula. Estos cambios causan a los alumnos ocasionales despistes que con el uso de esta aplicación sólo causarían pequeños retrasos en la llegada a la clase al conocerse donde se encuentran los compañeros de su grupo.

Habitualmente los alumnos quedamos para hacer trabajos en grupo o estudiar y no siempre es fácil que nos enteremos donde estamos ya que dependemos de la disponibilidad de las aulas. Además no siempre es fácil usar o entender el sistema de numeración estándar para las aulas de la UC3M sobretodo en los primeros años de estudio o si se visita otro campus.

1.1.3 Escenarios generales.

Aunque la aplicación se desarrolla en el entorno de la UC3M, de los escenarios anteriores es fácil extrapolar el problema a otros entornos como pueden ser:

- Congresos con numerosas salas donde se hacen distintas actividades al mismo tiempo. Especial relevancia tendrían los de tipo “Open space” ya que la agenda tiende a cambiar constantemente y se suelen generar espontáneamente nuevas actividades en lugares no previstos.
- Grandes factorías o centros de trabajo donde a veces la gestión y reserva de salas de reunión podría optimizarse con una gestión de presencia o en la necesidad de localizar a un empleado podría no resultar sencillo.
- Sistemas de fichada para el control de horas en el trabajo.
- Sistemas de presencia para optimizar rondas de seguridad.
- Centros comerciales. Tanto para que la tienda pueda saber que estas dentro o para que tus amigos o familiares con los que podías haber quedado sepan en que tienda estas.

1.2 Objetivo.

El principal objetivo perseguido ha sido desarrollar una aplicación llamada Find My Mates para ayudar a quienes deseen encontrar a personas en el ámbito de la UC3M haciendo uso de las capacidades de geolocalización y visualización de mapas tipo GIS de los smartphones.

Para ello la aplicación debe tener la capacidad de publicar su posición en un servicio externo al smartphone para que este pueda comunicar a quienes corresponda la posición compartida. Lo cual genera un objetivo secundario que consiste en desarrollar el servicio externo que permita compartir la ubicación de los usuarios. Las principales características del sistema son:

- Permitir al usuario ampliar el contacto que tenga con sus amigos de Facebook, ya que serán los contactos con los que podrá compartir la posición si son añadidos recíprocamente dentro de FMM.
- Los usuarios podrán definir en qué edificios se va a permitir al sistema que detecte y publique la ubicación de forma autónoma.
- El sistema proporcionará un listado de ubicaciones cercanas al usuario para que este haga una publicación manual de su posición.
- El sistema será capaz de mostrar las posiciones de los usuarios sobre un GIS.
- El sistema proporcionará al usuario la posibilidad de gestionar los contactos con los que se va a compartir la ubicación.
- El usuario debe poder elegir si el sistema comparte autónomamente la ubicación o no.

1.3 Metodología.

El método usado para llevar a cabo el desarrollo este proyecto ha sido iterativo. Las mejores características que tiene una metodología de este tipo son los ciclos de feedback y construcción incremental. Esta manera de actuar resulta beneficiosa ya que este proyecto se usa para experimentar con una tecnología nueva y los ciclos de feedback permiten validar hipótesis con las que se trabaja.

Al tratarse de un proyecto de aprendizaje y experimentación se desconoce a priori el ámbito y el alcance que se es capaz de cubrir por lo que se recomienda no tener un requisitos demasiado fijados desde un inicio pudiéndolos cambiar según se construye el sistema y según se van validando las hipótesis con el ciclo de feedback. Debido a la disponibilidad irregular de tiempo para dedicar al desarrollo de este proyecto, no es posible fijar un ciclo constante de tiempo para realizar una iteración. Por tanto el tiempo de ciclo vendrá determinado por el tiempo que se tarda en desarrollar y validar cada uno de los artefactos.

Los artefactos a desarrollar se encuentran en una pila de producto, por lo que para comenzar una iteración se elige uno de los artefactos con los que se va a trabajar. Cuando termina el trabajo se tiene que validar para pasar al siguiente artefacto. Con la validación se termina el ciclo y el resultado debe ser que un usuario puede disponer de una versión en la que sea posible usar todos los artefactos desarrollados en los ciclos anteriores.

Al ser un proyecto para aprender a desarrollar en un nuevo lenguaje, con nuevas herramientas para un nuevo entorno no se tiene fecha final de entrega y tampoco resulta útil tratar de adivinar cuanto se va a tardar en desarrollar cada artefacto en un primer momento. Aun así, si resulta de utilidad conocer el grado de avance del desarrollo y la velocidad de desarrollo para poder estimar en cada ciclo un momento de fin. Esta velocidad es un valor más que resulta útil para reevaluar requisitos si se quiere acortar el momento de fin.

Para lidiar con la situación anterior se evalúan los artefactos de la pila del producto. Para ello simplemente se comparan unos con otros para asignar un tamaño a cada uno en relación a los demás. Esto es, si se considera por cualquier razón que se estime en el momento de la evaluación que el desarrollo de un artefacto es de mayor tamaño que otro este tendrá mayor puntuación.

Con la pila de producto puntuada sabemos los puntos totales en ese momento que tiene el proyecto por lo que al finalizar cada ciclo podemos calcular una velocidad que resulta de dividir los puntos del artefacto desarrollados entre la unidad de tiempo que nos interese. Es en este momento, al final cada ciclo, cuando se puede observar el grado de avance y predecir un fin. Si cree conveniente se puede usar el conocimiento adquirido del desarrollo para reevaluar los puntos de los artefactos de la pila.

Para poder realizar el seguimiento de las tareas se utiliza una pizarra magnética en la que se pegan posits que representan los artefactos. Estos pasan de la pila de producto a una pila de finalizado. Este sistema muestra de forma rápida y visual el grado de avance del proyecto además de facilitar la priorización y visualizar un roadmap del trabajo pendiente.

Debido al gran tamaño de la pila de producto y del escaso espacio en la pizarra se usa de manera complementaria el issue tracking que ofrece Bitbucket junto al sistema de control de versiones del código.

Mi escasa experiencia con la metodologías de desarrollo no me permite asignar ninguna de las múltiples existentes a este proyecto pero me he basado en algunas ideas de Scrum, por tratar de implementar los principios de Agile, y de Kanban por sus conceptos de flujo de trabajo.

1.4 Contenido del documento.

Para transmitir al lector en que consta este documento se realiza una enumeración de cada uno de los capítulos con una pequeña descripción de su contenido.

Capítulo 1

Introducción al problema y los intereses que pretende cubrir el desarrollo de este proyecto

Capítulo 2

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Se hace un análisis de las tecnologías y las aplicaciones actuales que tienen cierta relación con el sistema que se va a desarrollar en este proyecto.

Capítulo 3

Se desarrolla el análisis de la aplicación. En este capítulo, y en base a los requisitos que definen lo que debe hacer el sistema, se definen nuevos requisitos para que estos puedan soportarse y unas historias de usuario que definirán la interacción del usuario con el sistema. Al terminar de leer el capítulo el lector conocerá de manera precisa la definición del sistema.

Capítulo 4

Si en el capítulo 3 se explicaba de forma precisa qué sistema se está construyendo en este se describe detalladamente cómo está construido. El lector conocerá en que componentes constructivos está dividido el sistema, cómo se han dividido y cómo interactúan entre ellos.

Capítulo 5

En este se definen las pruebas que debe pasar el sistema para que sea operativo y aceptado.

Capítulo 6

Este capítulo trata de explicar cómo se gestiona el proyecto desde su planificación hasta su desarrollo y entrega.

Capítulo 7

El proyecto deja líneas abiertas por lo que en este capítulo se trata de dar unas conclusiones, explicar las líneas a seguir para continuar el desarrollo y sugerir nuevas características o mejoras que podrían ser interesantes para mantener vivo el proyecto.

1.5 Referencias

Lean software development: An agile toolkit

Mary Poppendieck y Tom Poppendieck

Lean from the trenches: Managing Large-Scale Projects with Kanban

Henrik Kniberg

The agile samurai: How Agile Masters Deliver Great Software

Jonathan Rasmusson

Find My Mates

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

The official Scrum Guide

Jeff Sutherland, Ken Schwaber

Capítulo 2: Análisis del estado de la cuestión

Contenido

2	ANÁLISIS DEL ESTADO DE LA CUESTIÓN.....	18
2.1	APLICACIONES SIMILARES.....	18
2.1.1	<i>Whatsapp.....</i>	<i>18</i>
2.1.2	<i>Buscar a mis amigos.....</i>	<i>19</i>
2.1.3	<i>Google Maps.....</i>	<i>20</i>
2.1.4	<i>UC3M Mapas.....</i>	<i>21</i>
2.1.5	<i>Waze.....</i>	<i>22</i>
2.1.6	<i>Google+.....</i>	<i>22</i>
2.1.7	<i>Facebook.....</i>	<i>23</i>
2.1.8	<i>Foursquare.....</i>	<i>24</i>
2.1.9	<i>Swarm.....</i>	<i>25</i>
2.1.10	<i>Comparativas y conclusiones.....</i>	<i>26</i>
2.2	ANÁLISIS DE TECNOLOGÍAS DISPONIBLES.....	28
2.2.1	<i>Heroku.....</i>	<i>28</i>
2.2.2	<i>Amazon AWS.....</i>	<i>28</i>
2.2.3	<i>Google cloud.....</i>	<i>29</i>
2.2.4	<i>Comparativas y conclusiones.....</i>	<i>29</i>

2 Análisis del estado de la cuestión.

Para el desarrollo de este proyecto se hace un breve análisis sobre las aplicaciones existentes que podrían tener funcionalidades u objetivos similares que pueden ayudarnos a resolver el problema detectado o que nos inspiran para el desarrollo. Si bien existen aplicaciones que pueden ayudarnos a resolver nuestros problemas, encontramos que tienen una serie de carencias por lo que se decide desarrollar una nueva aplicación.

También para el desarrollo será necesario analizar los distintos entornos y tecnologías disponibles que nos pueden ayudar a soportar nuestra idea de producto.

2.1 Aplicaciones similares.

Se ha hecho una selección de aplicaciones de mensajería, de mapas, sociales y de navegación por GPS que contienen características y funcionalidades similares a las funcionalidades que contiene el sistema que se quiere desarrollar.

Se consideran similares aquellas aplicaciones que tengan algún caso de uso en el que el usuario pueda publicar el lugar en el que se encuentra y el usuario pueda ver donde se encuentran sus contactos. También se consideran similares si estos lugares están geo-referenciados y se puede ver su situación geográfica en un GIS. Otro caso de uso que se ve de utilidad en las aplicaciones similares es que el usuario pueda elegir el lugar en el que se encuentra para publicarlo a partir de unas sugerencias del sistema.

2.1.1 Whatsapp

Esta es una aplicación de mensajería con la que los usuarios pueden mantener conversaciones de texto con sus contactos. Las aplicaciones de mensajería modernas como Whatsapp también dan la posibilidad de compartir la posición de los usuarios en las conversaciones.

La única aplicación de las analizadas de mensajería que usa la api de google places es Whatsapp, por lo que merece una mención particular al aportar mayor semántica o conocimiento a la ubicación compartida.

Esta aplicación se ha escogido porque permite compartir la ubicación del usuario con sus contactos y por contener una sección en la que se listan una serie de lugares catalogados que se encuentran cercanos al usuario para que este envíe su ubicación. Esta última será especialmente útil en el sistema en desarrollo para que el usuario publique su emplazamiento en los interiores de los edificios de la UC3M usando el catálogo de ubicaciones de la universidad en lugar del de Google Places.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Ilustración 1: Uso de la API de Google Places en Whatsapp

2.1.2 Buscar a mis amigos.

<https://www.apple.com/es/apps/find-my-friends/>

Amigos es una aplicación de Apple para iOS que te permite ver la posición de los contactos del usuario sobre un GIS. Además tiene la posibilidad de agregar notificaciones para cuando el contacto llegue o se vaya de un lugar concreto.

Para que esto sea posible es necesario que el usuario mande una invitación al contacto y que este la acepte. A partir de este momento el sistema comienza a hacer un seguimiento del contacto.

Esta es una aplicación de carácter social que se ha escogido por mostrar la posición de los contactos en un GIS. Esta característica resulta útil para el sistema de este proyecto porque ayudaría al usuario a conocer las posiciones de los contactos si no se conocen los nombres de los edificios o la nomenclatura numerada que determinan los emplazamientos dentro de la universidad.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

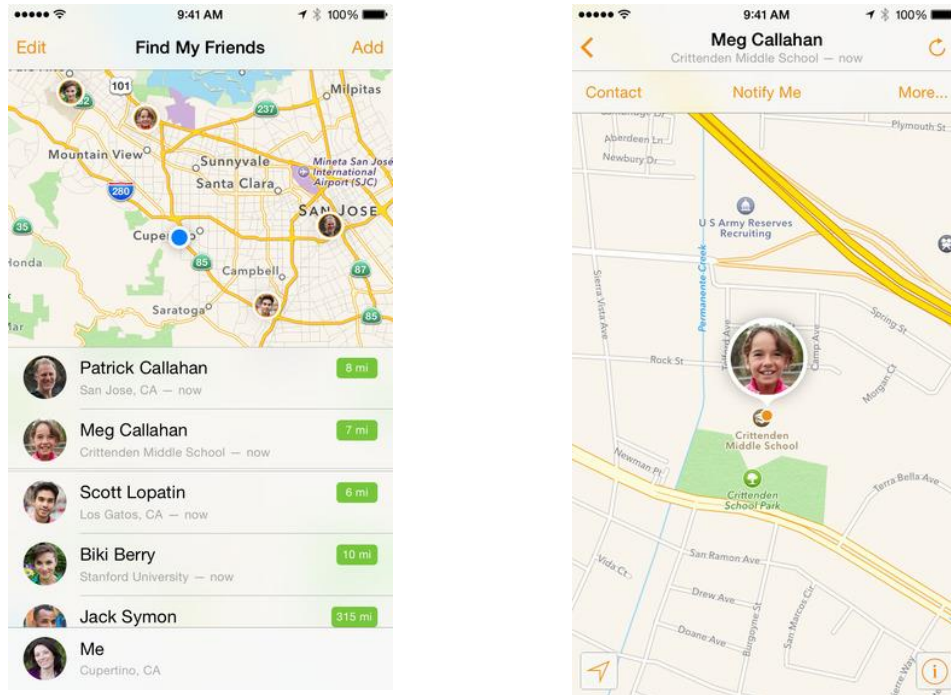


Ilustración 2: Interfaz de usuario de Busca mis amigos.

2.1.3 Google Maps

<http://maps.google.es>

Este sistema, al igual que [Bing Maps](#), incorporan un GIS y ayudan a localizar direcciones o lugares. El valor que añade Google Maps es el contener mayor cantidad de información para los usuarios y ofrece mayores facilidades para que estos puedan añadir o modificar información relativa a los lugares, como la corrección de la localización de un establecimiento. También es capaz de mostrar comentarios, puntuaciones y comentarios sobre los lugares consultados.

Esta es una aplicación de mapas que muestra la posición del usuario en un GIS y le ofrece información sobre sus alrededores. Situar al usuario en un mapa en este proyecto se ve necesario para situarlo con respecto a sus contactos y al entorno que le rodea.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

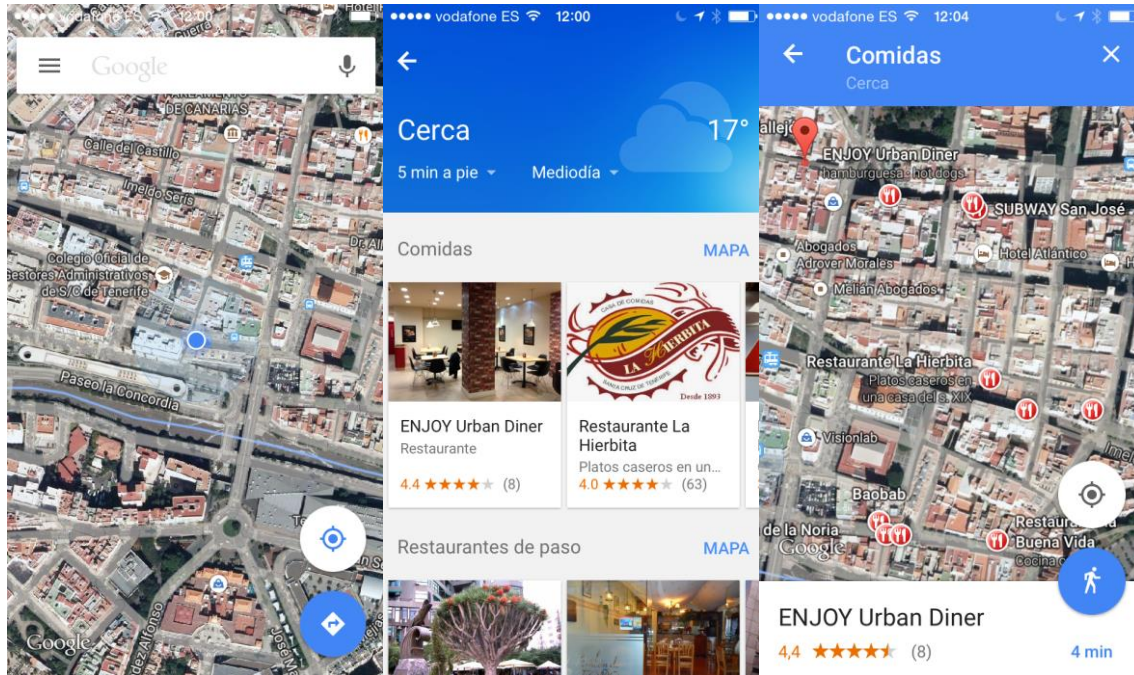


Ilustración 3: Google Maps mostrando sugerencias de lugares que son geo-referenciados

2.1.4 UC3M Mapas

<https://itunes.apple.com/us/app/uc3m-mapas/id515265440?l=es&ls=1&mt=8>

Es una aplicación de mapas desarrollada por la UC3M que da acceso al catálogo de emplazamientos de la universidad.

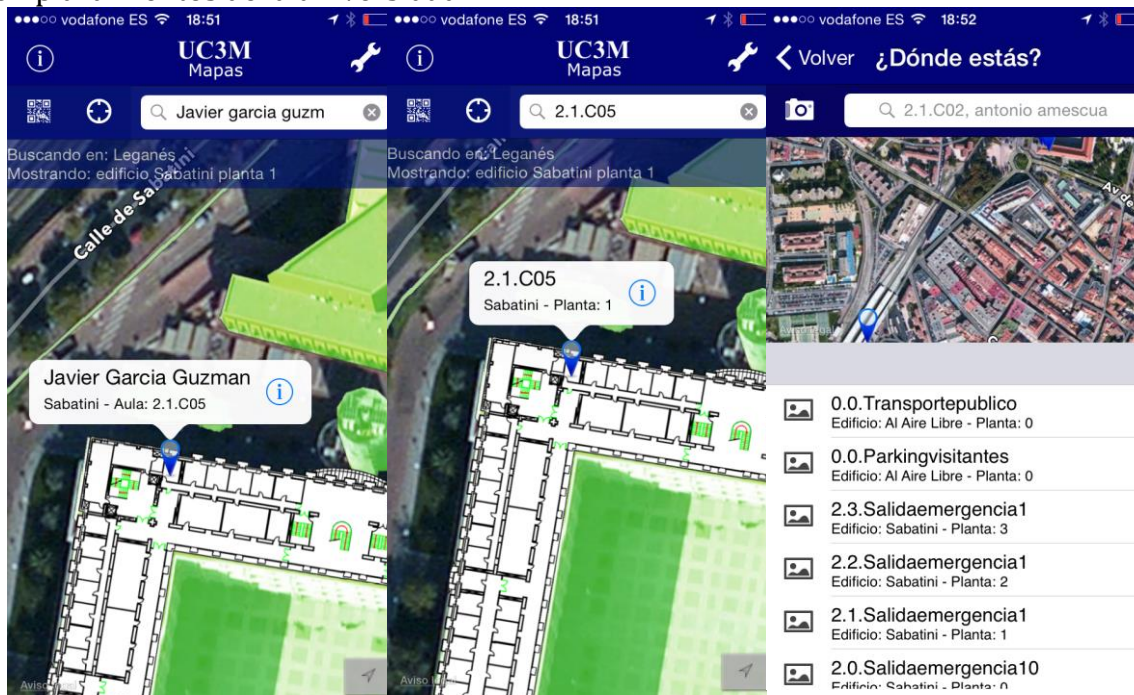


Ilustración 4: Búsqueda de personas, ubicaciones directas y sugerencias de ubicación a buscar

Una de las características interesantes de la aplicación es que relaciona los nombres de las localizaciones de la universidad con su situación geográfica añadiendo un plano de distribución del interior de los edificios, aunque a día de hoy google maps muestra la distribución interna de la universidad sin necesidad de añadir capas adicionales desde la aplicación o a google maps engine. Además ofrece una relación

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

entre el personal de la UC3M con sus ubicaciones por defecto para conocer, por ejemplo, cual es el despacho de algún profesor.

Es en esencia una aplicación para encontrar lugares dentro de la universidad. Con lo que se permite al usuario hacer búsquedas de lugares para saber como llegar o de personas para conocer su ubicación por defecto como puede ser su despacho, aunque esto no implica saber si se encuentra en el o no. Es por tanto esta la deficiencia principal que pretende salvar el sistema en desarrollo.

El sistema de búsqueda también permite al usuario encontrar emplazamientos de la universidad cercanos, lo cual puede ser usado por Find My Mates para publicar la posición del usuario dentro de la universidad.

2.1.5 Waze

<https://www.waze.com/>

Es un navegador GPS con una importante componente social. Este te permite no solo conocer la posición en tiempo real de los usuarios e interactuar con ellos sino que además se puede usar la información correspondiente a la ruta que se sigue que aportan el resto de usuarios del sistema.

Se ofrece la posibilidad de navegar a los eventos de Facebook y ver qué amigos se están dirigiendo al destino conociendo su tiempo estimado de llegada.



Ilustración 5: Interfaz de la parte social de Waze.

Esta aplicación se ha escogido por tener la característica de mostrar las posiciones de los contactos de Facebook en un mapa si estos dan su consentimiento. El proyecto que se está desarrollando pretende usar los contactos de Facebook para identificar a los usuarios.

2.1.6 Google+

<http://plus.google.com>

Google+ permite compartir tu posición en tiempo real con los contactos que desees. Incluso un historial de ubicaciones. Permite dos niveles de precisión para compartir la ubicación. La ubicación exacta permite que los contactos puedan ir a donde te

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

encuentras y la ubicación por ciudad solo permite a los contactos que elige el usuario conocer en qué ciudad se encuentra.

También permite hacer publicaciones en el muro o timeline de posición. Esta es elegida de un listado con ubicaciones próximas sugeridas y al igual que la anterior puede ser abierta en Google Maps por tus contactos para saber cómo llegar a tu posición o conocerla mejor.

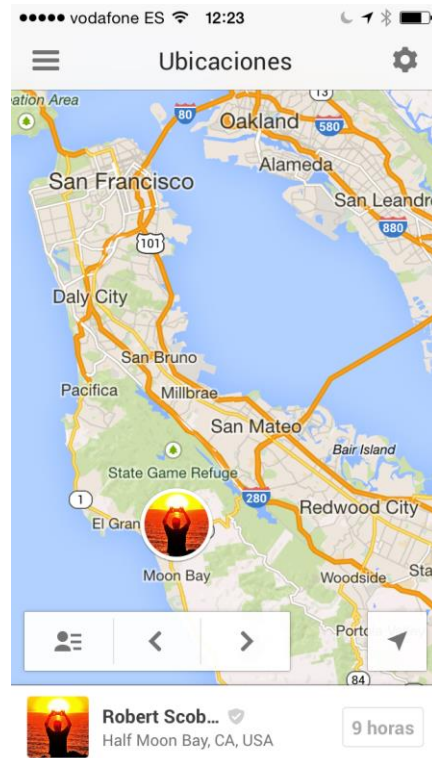


Ilustración 6: Interfaz de ubicaciones de Google+

Esta aplicación es de carácter social y como se ve en la captura muestra la posición de los contactos sobre el GIS de Google. Este GIS tiene entorno de desarrollo para numerosas plataformas, por lo que merece la pena su uso al ser aprovechable el conocimiento que se obtenga de él para otros proyectos en otros entornos.

2.1.7 Facebook

www.facebook.com

Los usuarios de Facebook pueden publicar su posición con los amigos que deseen. Esta es elegida de un listado que sugiere ubicaciones próximas y los contactos con la que la compartas pueden abrirla en Mapas para saber cómo llegar a la posición publicada. También se puede buscar información sobre los lugares cercanos. Esta puede contener comentario de tus amigos si han estado y puntuación general que le dan los usuarios.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

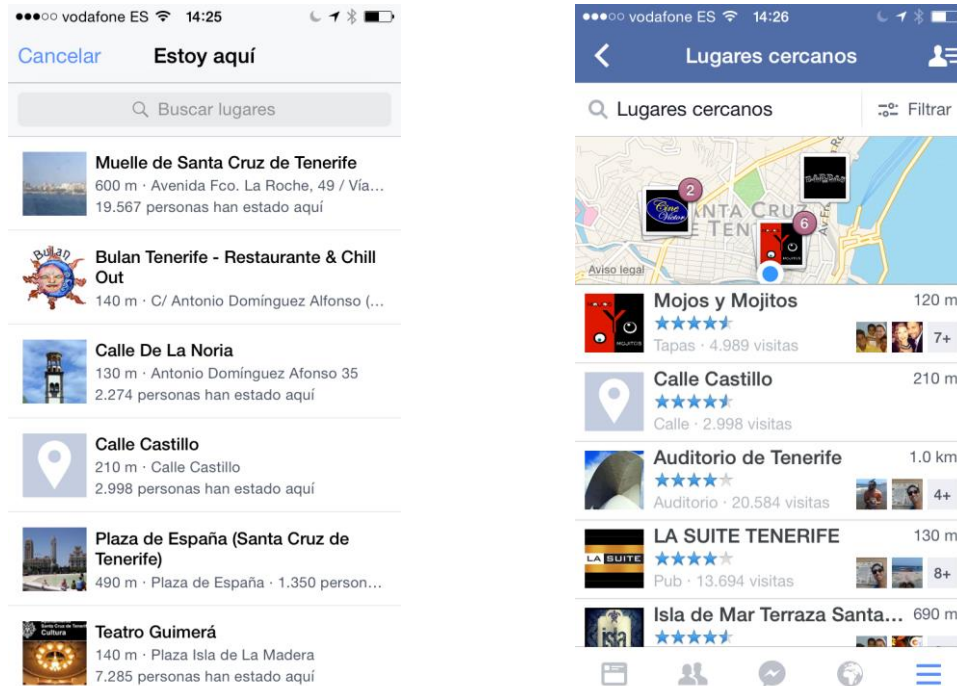


Ilustración 7: Interfaces de Facebook en iOS en la que se muestran lugares según posición del usuario

Esta aplicación, al igual que Google+ es de carácter social. Es una aplicación interesante al tener una función que permite al usuario consultar que emplazamientos se encuentran cerca del él. También tiene un GIS, esta vez el de Apple pero este componente no es reutilizable en otros proyectos en un entorno que no sea el de su propietario.

2.1.8 Foursquare

<http://www.foursquare.com>

Foursquare pretende aprender de tus gustos y el de tus contactos para ayudarte a encontrar lugares que puedan interesarte.

Clasifica los lugares en comida, café, vida nocturna, compras, aire libre y entretenimiento. La manera que tiene de aprender y ayudar a los usuarios a encontrar lugares es mediante los checkins que hacen los usuarios manualmente que pueden incluir etiquetas sobre lo que han consumido, comentarios, precios o puntuación que se le da al lugar.

Versiones anteriores permitían hacer checkin en lugares y añadir cualquier tipo de comentario al respecto. En las versiones actuales se permite añadir fotos y comentarios del lugar pero la funcionalidad de checkin ha quedado delegada en la aplicación swarm que se comenta en el siguiente apartado. La funcionalidad que permite añadir comentarios de los lugares puede resultar útil para los usuarios del sistema en desarrollo si quisieran aportar información, por ejemplo, sobre la hora hasta la que esperan estar disponibles en ese lugar.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

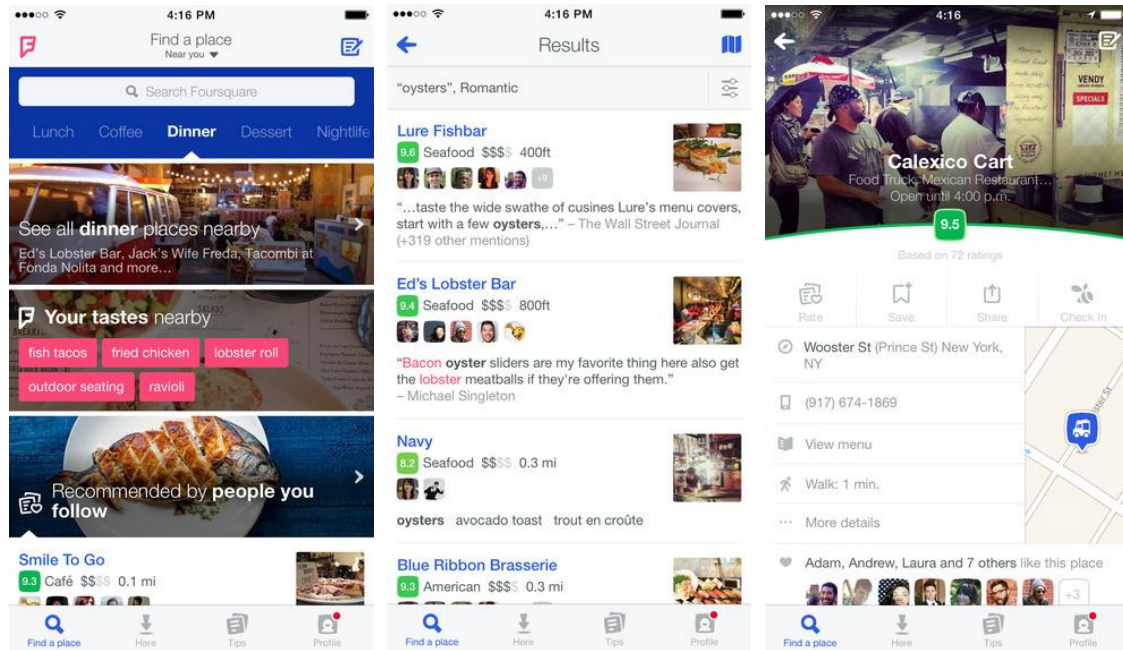


Ilustración 8: Foursquare donde ofrece lugares y posibilidad de hacer checkin.

2.1.9 Swarm

<https://www.swarmapp.com/>

Está desarrollada por Foursquare Inc. Permite hacer checkin un lugar sugerido o compartir de manera autónoma el barrio en el que se encuentra el usuario.

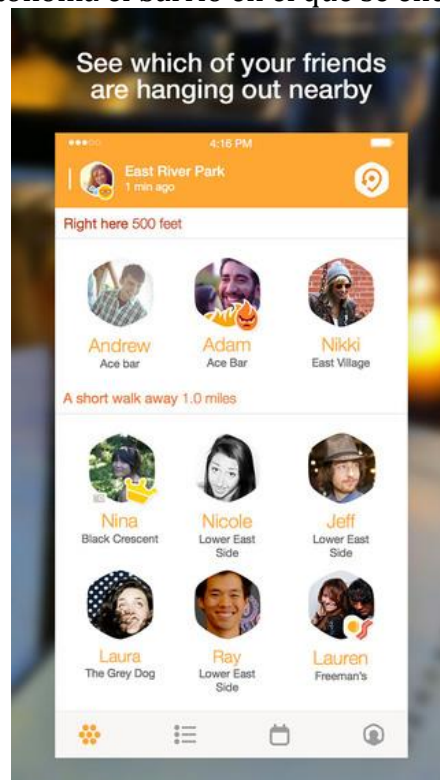


Ilustración 9: Interfaz de Swarm mostrando lugares en los que se encuentran los contactos del usuario.

Esta aplicación ha sido incluida en el análisis por tener un uso enfocado en la publicación de la ubicación de los usuarios resultando interesante la visualización de las posiciones de los contactos ordenadas por distancias.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

2.1.10 Comparativas y conclusiones

Las aplicaciones analizadas se comparan según unas características que se definen a continuación:

- **Compartir ubicación de forma autónoma.** Se define como la capacidad de la aplicación para compartir o publicar la posición sin interacción por parte del usuario.
- **Compartir ubicación de forma manual.** Se define como la capacidad de la aplicación para compartir o publicar una posición indicada por el usuario
- **Sugerencia de sitios.** Se define como la capacidad de la aplicación para sugerir lugares en función de la posición leída por los sensores del dispositivo en el momento de ser sugeridos.
- **Usa eventos de geofencing.** Se define como la capacidad de la aplicación para detectar cuando se cruza un área determinada
- **Gestión de usuarios** es la capacidad de la aplicación para añadir y eliminar usuarios con los que se desea compartir la información
- **Gestión de lugares** es la capacidad que tiene la aplicación para añadir o eliminar lugares en los que se desea compartir la posición de manera autónoma.
- **Mostrar posición de contactos sobre un GIS** implica que la aplicación contiene un GIS en el que pueden señalarse la posición de todos los contactos que la han compartido.
- **Posición en interiores** se refiere a que la aplicación tiene la capacidad de anunciar que el usuario se encuentra en el interior de algún emplazamiento.
- **Conoce el directorio de la UC3M.** Es la información necesaria para saber cuál es el despacho de un profesor o donde se encuentra un aula determinada.
- **Publicación continua de la posición.** Se refiere a que la aplicación está continuamente publicando la posición del dispositivo dando la sensación de movimiento a quien observa o haciendo posible trazar el camino exacto seguido por el dispositivo.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

	WhatsApp	Amigos	Google Maps	UC3M Mapas	Google+	Facebook	Foursquare	Swarm	Waze	Find My Mates
Compartir ubicación de forma autónoma	✗	✓	✗	✗	✓	✗	✗	✓	✓	✓
Compartir ubicación de forma manual	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓
Sugerencias de sitios	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓
Usa eventos de geofencing	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
Gestión de usuarios	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓
Gestión de lugares	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Mostrar posición de contactos en un GIS	✗	✓	✓	✗	✗	✓	✗	✓	✓	✓
Posición en interiores	✗	✓	✗	✗	✗	✗	✓	✓	✗	✓
Conoce el directorio de la UC3M	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Enlazado con redes sociales	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Publicación continua de la posición	✗	✓	✗	✗	✓	✗	✗	✗	✓	✗

Tabla 1: Comparativa de aplicaciones

Para orientar el desarrollo de Find My Mates se han escogido virtudes de las aplicaciones analizadas y deseado características que no se consideran apropiadas para los objetivos que persigue el proyecto.

Compartir una ubicación de forma autónoma se considera una virtud. Al usar los eventos de geofencing que puede controlar el dispositivo se puede publicar sin que el usuario actúe la posición del dispositivo. Para ello se ven otras dos virtudes como la gestión de lugares para decidir cuáles son las áreas de geofencing a las que va a responder nuestra aplicación y la gestión de usuarios para definir los usuarios que tienen permiso para ver nuestra posición.

La sugerencia de sitios se ve esencial a la hora de hacer un checkin manual, ya que ofrece posiciones informadas en el interior de edificios donde es difícil tener una cobertura nítida de GPS y emitir una publicación veraz sobre el posicionamiento del usuario. Estos sitios sugeridos pertenecen al directorio de la UC3M por lo que se ve

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

de gran utilidad que la aplicación pueda conectarse al mismo y obtener información georeferenciada de las posiciones interiores de los edificios o cercanas al usuario.

El enlace con las redes sociales se ve de utilidad para ser usada como base de datos de usuarios y control de acceso a la aplicación.

El caso de la publicación continua de la posición no se ve como una cualidad que deba tener Find My Mates para aportar algo más de privacidad y porque tampoco se ve necesario para acometer los objetivos del proyecto.

2.2 Análisis de tecnologías disponibles.

Para que todos los usuarios puedan comunicarse las aplicaciones deben conectarse a un servicio externo al dispositivo de cada usuario. Es por esto por lo que se analizan algunas plataformas que permitan el desarrollo de dicho servicio. En esencia se necesita desarrollar un servicio REST que nos permita almacenar y recuperar información generada por Find My Mates y que implemente algunas reglas de negocio que gestionen la privacidad de los usuarios.

Las tecnologías analizadas están desarrolladas y orientadas a ser usadas principalmente por desarrolladores de software y presumen de ahorrar a estos costes de mantenimiento y gestión de servidores.

2.2.1 Heroku

<https://www.heroku.com/>

Es una plataforma como servicio (PAAS) en la nube que soporta varios lenguajes de programación para ser ejecutados en ella.

La plataforma se puede configurar en función de las necesidades de procesamiento, configurando Dynos, y en función de las necesidades de almacenamiento configurando los Heroku Postgres. Es la plataforma sugerida por Facebook y permite desplegar aplicaciones desarrolladas en ruby, php, node.js, python, java, clojure y scala.

Tiene un sistema de Add-ons que permite usar tecnologías más específicas como por ejemplo añadir un almacén de datos redis, caches, sistemas de login o monitorización.

Ofrece al usuario un sistema de cuotas, tanto de procesamiento como de almacenamiento que permiten escalar la plataforma en función de los requerimientos de la aplicación desplegada. Además los add-ons pueden tener su propio sistema de cuotas.

Normalmente hay una cuota de uso gratuita con un límite a partir del cual se debe pagar por el servicio.

2.2.2 Amazon AWS

<https://aws.amazon.com/es/>

Es una colección de servicios de computación remota. Como heroku presume de proveer de un servicio con altas capacidades de procesamiento y una plataforma de alta escalabilidad. Ofrece más un servicio de infraestructura que de plataformas pudiendo tener instancias de máquinas con Linux o Windows a las que se le pueden añadir distintos servicios a través de un market place.

Tiene dos productos principales, EC2 para procesamiento y S3 para almacenamiento. EC2 ofrece 750 horas gratuitas en sus instancias de máquinas más sencillas y S3 ofrece 5GB, 20000 operaciones de lectura y 2000 de escritura de forma gratuita.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Disponen de SDK para Android, iOS, Javascript, java, php, .NET, Python, Node.js y Ruby

2.2.3 Google cloud

<https://cloud.google.com/>

Es la plataforma de computación en la nube de Google.

Los grandes productos son Compute Engine y App Engine. El primero basado en soluciones IAAS y la segunda en PAAS.

App Engine ofrece un escalado automático de la plataforma para manejar la carga del servidor. Este escalado puede ser manual si se quiere tener mayor control en la facturación. Es el servicio que más se parece a Heroku y soporta los lenguajes Java, Python, Go y PHP.

Compute Engine. Ofrece al usuario tener distintas instancias de máquinas virtuales que pueden ser instanciadas bajo demanda. Es el servicio que más se parece a Amazon WS aunque este parece ofrecer un escalado más sencillo y dinámico.

Como los otros servicios vistos también ofrece cuotas de uso gratuito

2.2.4 Comparativas y conclusiones

Amazon AWS es la solución más compleja. Al ser un servicio de infraestructura se requiere mayor conocimiento de las capacidades de las máquinas que se usan y requiere de hacer una instalación de las plataformas que se estimen adecuadas para el sistema junto con hacer la configuración y posterior monitorización y pruebas para asegurar que la plataforma está instalada como se quiere y funciona de manera óptima. Esta flexibilidad y control que se ofrece da la facilidad a largo plazo de reutilización de código en migraciones al basarse en plataformas conocidas pero el coste de bootstrapping es mayor que usando un servicio de plataformas que abstraiga a los desarrolladores de la infraestructura que la sostiene.

Heroku es mucho más sencillo de configurar al ser un servicio de plataforma. Según lo que se observa en la documentación, desarrollar y desplegar una aplicación para que se ejecute en Heroku parece ser más rápido y sencillo que para Amazon AWS. Como tiene menos opciones probablemente como solución a largo plazo no sea la mejor al no permitir todas las optimizaciones que podrías tener en Amazon WS.

De las soluciones de Google me decanto por App Engine. Al ser, como Heroku, una tecnología de PAAS abstrae a los desarrolladores de la infraestructura del sistema. En Google App Engine no es obligatorio configurar una base de datos, al contrario que con heroku, ni máquinas y de entrada permite partir de una base muy completa aunque no esté basada en soluciones conocidas y comerciales. Esta solución base tiene la desventaja de no poder reutilizar la capa de presentación y la de datos en caso de querer usar otras soluciones o librerías pero tiene la gran ventaja de la simpleza y el corto tiempo que se tarda en desplegar por primera vez una aplicación. Es por eso por lo que se ha preferido usar en este proyecto App Engine en lugar de las otras opciones

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma
IOS)

Capítulo 3: Especificación de requisitos

Contenido

3	ESPECIFICACIÓN DE REQUISITOS SOFTWARE.....	32
3.1	INTRODUCCIÓN.....	32
3.1.1	Propósito.....	32
3.1.2	Ámbito.....	32
3.1.3	Definiciones, Acrónimos y Abreviaturas.....	32
3.1.4	Referencias.....	33
3.1.5	Visión general del documento.....	33
3.2	DESCRIPCIÓN GENERAL.....	33
3.2.1	Perspectiva del producto.....	33
3.2.2	Funciones del producto.....	35
3.2.3	Características de los usuarios.....	36
3.2.4	Restricciones.....	36
3.2.5	Suposiciones y dependencias.....	36
3.2.6	Requisitos futuros.....	36
3.3	REQUISITOS ESPECÍFICOS.....	37
3.3.1	Interfaces externas.....	37
3.3.2	Funciones.....	39
3.3.3	Requisitos de rendimiento.....	46
3.3.4	Restricciones de diseño.....	47
3.3.5	Atributos del sistema.....	47

3 Especificación de requisitos software.

3.1 Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS o en inglés SRS). Consta de varias subsecciones:

- Propósito. Aquí se expresa el propósito del documento ERS y a quien va dirigido
- Ámbito. Donde se da nombre al sistema y se explica lo que hará
- Definiciones, Acrónimos y abreviaturas para definir los términos usados en este documento
- Referencias para listar los documentos usados en la ERS
- Visión general para describir el contenido del resto del ERS

3.1.1 Propósito.

El propósito del presente documento es presentar una descripción detallada del sistema que se va a desarrollar. Va dirigido a todos los participantes en el desarrollo del producto para poner en conocimiento y entablar las bases de lo que el sistema debe hacer, incluyendo en los participantes al departamento de informática de la Universidad Carlos III de Madrid para su aprobación.

3.1.2 Ámbito.

La solución software que se desarrollará será un sistema para la geo-localización de personas en grandes edificios o grandes centros tipo campus llamada Find My Mates (FMM). Concretamente se desarrolla para la Universidad Carlos III de Madrid y para que sea usado dentro del ámbito universitario en los centros de la universidad.

Find My Mates es una aplicación que se desarrolla para dispositivos iPhone. Es capaz de compartir la ubicación del usuario con el resto de usuarios de la aplicación.

El acto de publicar una ubicación se denomina en el contexto del sistema como checkin. Este puede ser automático, para publicar el edificio en el que se encuentra el usuario, o manual, para dar la posibilidad al usuario de que publique una posición más concreta dentro del edificio en el que se encuentre.

3.1.3 Definiciones, Acrónimos y Abreviaturas.

Checkin	Acción de publicar una posición puede ser manual o automática.
---------	--

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

3.1.4 Referencias.

IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

3.1.5 Visión general del documento.

El siguiente apartado de este documento, Descripción General, ofrece una visión general de la funcionalidad del producto. Describen los requisitos y será usado para establecer el contexto para los requisitos técnicos del apartado que le suceda.

El tercer apartado, Especificación de requisitos va dirigido principalmente a los desarrolladores y describirá detalladamente la funcionalidad del producto.

3.2 Descripción general.

3.2.1 Perspectiva del producto.

La aplicación que se desarrolla se llama Find My Mates. Es una aplicación para encontrar personas en la Universidad Carlos III de Madrid (UC3M). Desarrollada para iPhone facilita a los usuarios publicar en qué lugar de la UC3M se encuentran y así poder ser encontrados.

Existen dos modos de publicación: autónoma y manual. La autónoma consiste en que el sistema es capaz de detectar en qué edificio se encuentra el usuario y lo publica sin necesidad de que el usuario interactúe en ese momento con la aplicación, pero previamente ha tenido que añadir el edificio a la lista de edificios de autocheckin y dar permiso a la aplicación para que ejecute esta acción de forma autónoma. La publicación manual está pensada para cuando un usuario se encuentra en un aula, o despacho de la universidad y quiere publicarlo. En este caso y según las coordenadas del usuario, la aplicación muestra un listado de emplazamientos cercanos a sus coordenadas que el usuario puede seleccionar para hacer el checkin.

La solución que se propone consiste en una aplicación desarrollada para iPhone. Esta se apoya en un servicio web que se ejecuta en el App Engine de Google para el almacenamiento de los checkins de los usuarios.

Los usuarios se identifican como usuarios de Facebook de manera que se puede aprovechar su sistema de autenticación.

Para poder mostrar las posiciones de los usuarios en un mapa se usa el SDK de Google Maps para integrar su GIS en la aplicación.

Por otra parte, como los usuarios publican lugares dentro de la UC3M, se usa su catálogo de emplazamientos geolocalizados mediante el uso de la API de UC3M Mapas que permitirá a la aplicación relacionar unas coordenadas geográficas con el nombre del lugar en el que se encuentra el usuario.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

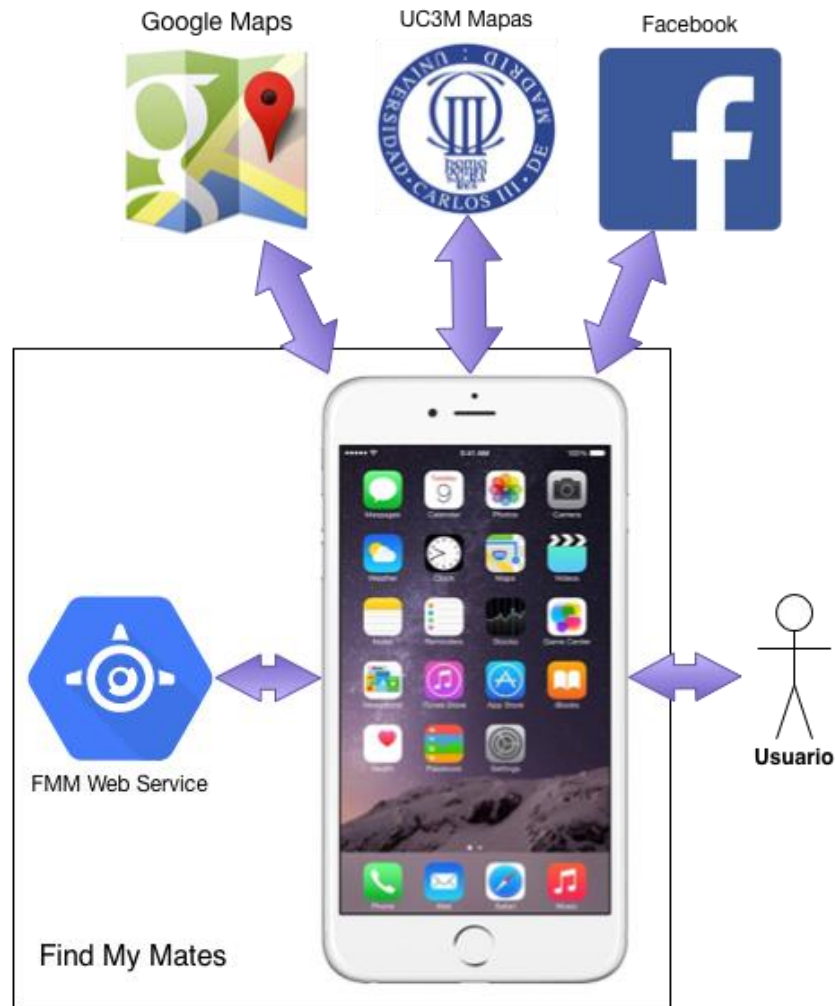


Ilustración 10: Perspectiva del producto

3.2.1.1 Interfaces externas

3.2.1.1.1 Google Maps

La interfaz con Google Maps se materializa haciendo uso de su SDK para iOS. Este se encarga de solicitar al servicio de Google Maps la imagen que representa el mapa en función de la localización geográfica de la cámara y el zoom.

3.2.1.1.2 UC3M Mapas

Esta interfaz es accesible mediante una API REST mediante la cual Find My Mates es capaz de obtener los campus de la UC3M, los edificios que comprenden los campus y buscar las localizaciones cercanas a una posición geográfica dada.

3.2.1.2 Interfaces de usuario.

El usuario dispone de cuatro interfaces principales a las que puede acceder mediante el uso de una barra de pestañas.

3.2.1.2.1 Contactos

En esta pantalla se muestra el listado de contactos con los que se desea compartir la ubicación. Los contactos en esta pantalla han sido agregados al sistema haciendo una búsqueda y posterior selección de los amigos que tiene el usuario en Facebook.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Esta pantalla da acceso a la visualización de los contactos sobre el GIS de Google Maps y a la pantalla que permite añadir contactos del Facebook con los que el usuario desea compartir su ubicación.

3.2.1.2.2 Edificios

En esta pantalla el usuario puede ver un listado con los edificios en los que el sistema detectará su presencia y la publicará automáticamente a sus contactos.

En esta pantalla también se permite añadir nuevos edificios que son proporcionados por el servicio web de mapas de la universidad

3.2.1.2.3 Checkin

Esta pantalla está destinada a que el usuario haga checkin de forma manual. Lista lugares cercanos al usuario en los que puede hacer un checkin manual y publicar de esta manera su ubicación a sus contactos.

3.2.1.2.4 Ajustes

En esta pantalla el usuario tiene la opción de activar o desactivar el autocheckin y de hacer login con su cuenta de Facebook.

3.2.2 Funciones del producto.

A continuación en este apartado se describen las funciones que deberá tener la solución propuesta para que sea aceptada.

3.2.2.1 Gestión de contactos

Los usuarios podrán añadir amigos de Facebook para que sean contactos de la aplicación permitiendo así que estos conozcan la posición del usuario.

También los usuarios deben poder eliminar contactos con los que ya no se desee compartir la ubicación.

3.2.2.2 Gestión de edificios

Los usuarios podrán añadir edificios de la UC3M para que el sistema publique a los contactos automáticamente cuando el usuario se encuentre en el interior.

También los usuarios podrán eliminar edificios para que el sistema deje de publicar a los contactos cuando el usuario se encuentre en el interior.

3.2.2.3 Checkin

Al acto de publicar la posición del usuario se le llama checkin y la Find My Mates tiene dos modos para ello.

Un modo autónomo se da cuando el usuario entra en un edificio. Si el usuario previamente ha añadido el edificio en el que entra, el sistema detecta cuando se encuentra en el interior y lo publica a los contactos.

El checkin manual se hace mediante una interfaz de usuario que, en función de las coordenadas del usuario, muestra un listado de lugares cercanos que se pueden seleccionar para publicar la posición.

3.2.2.4 Ajustes

Los usuarios podrán configurar si quieren que la aplicación publique de forma autónoma en los edificios que se entra.

También podrán iniciar sesión con su cuenta de Facebook o cerrarla.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

3.2.3 Características de los usuarios.

Los usuarios de la aplicación deben ser usuarios de Facebook. Los usuarios que entiendan lo que es Facebook, como funciona y en que consiste no tendrán ningún inconveniente en usar Find My Mates.

Existe un usuario especial para el uso de la consola de administración de Google App Engine. Este además de tener capacidades técnicas avanzadas para la administración de sistemas debe conocer el lenguaje de script python.

3.2.4 Restricciones.

Para fijar un alcance se enumeran a continuación una serie de restricciones del sistema:

- El sistema se desarrolla para iOS 8.1
- Para la comunicación con Facebook se hace uso de FBS DK 3.20
- La API de Facebook usada es FB Graph API 2.2
- El SDK usado para mostrar un GIS en la aplicación es el de Google Maps versión 1.9
- El iPhone para el que se desarrolla esta aplicación debe tener la capacidad de estar permanentemente conectado a internet.

3.2.5 Suposiciones y dependencias.

El correcto funcionamiento de Find My Mates depende de sistemas externos como son la API de Google Maps, Facebook, y el servicio web de Mapas de la UC3M. Si estos no funcionan como se espera Find My Mates tendrá un comportamiento también inesperado.

Estos servicios muestran cierta fiabilidad por lo que las causas más probables de fallo sea un cambio en la definición de las interfaces con las que permiten que otras aplicaciones se comuniquen con ellos o la falta de conectividad a internet.

También tienen dependencias internas. El SDK de Google Maps, y el Facebook pueden dejar de funcionar correctamente si se actualiza el sistema operativo del dispositivo en el que se ejecutan o cambian las APIs de comunicación y no se actualiza el SDK.

La solución propuesta se apoya en el supuesto de que el dispositivo en el que se ejecuta se encuentra conectado a internet. En un entorno sin conectividad FMM no puede cumplir con la satisfactoria ejecución de los casos de uso propuestos.

3.2.6 Requisitos futuros.

Identificador	Nombre
RF-13	Auto-checkin interno

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Descripción

El sistema debe tener la capacidad de hacer checkin de manera autónoma en las distintas estancias interiores de los edificios

Identificador

RF-14

Nombre

QR Checkin

Descripción

El sistema debe tener la capacidad de hacer checkin manual usando la cámara del teléfono sobre códigos QR que identifican las estancias interiores de los edificios

3.3 Requisitos específicos.

3.3.1 Interfaces externas.

En este apartado solo se detalla la interfaz con UC3M Mapas ya que las interfaces con Google Maps y con Facebook quedan ocultas por los SDK de los proveedores del servicio que se usan.

3.3.1.1 UC3M Mapas

Las funciones de la API que resultan útiles son: Campuses, Buildings y Find. Que permiten al desarrollador obtener respectivamente los campus que hay en la UC3M, los edificios por campus y buscar las estancias cercanas. A continuación se enumeran las funciones sin mencionar los detalles, que deberán ser consultados en la documentación de la API

3.3.1.1.1 Campuses

Esta función permite obtener los campus que existen en la UC3M

3.3.1.1.2 Buildings

Esta función permite obtener los edificios que hay en un campus pasado por parámetros

3.3.1.1.3 Find

Esta función devuelve las localizaciones cercanas a unas coordenadas que pueden ser pasadas por parámetros.

3.3.2 Interfaces de usuario

La aplicación consiste en cuatro vistas principales organizadas en un control de pestañas.

3.3.2.1 Contacts

Esta pestaña consiste en un control de navegación con vistas que permite transitar a la vista de búsqueda de contactos o a la vista del mapa.

La vista principal consiste en una vista de tabla en la que se muestran en celdas cada contacto que el usuario a añadido.

Las celdas se dividen en tres líneas y muestran los siguientes campos:

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Nombre:	Es el nombre con el que el contacto aparece en Facebook. Este campo es obligatorio y se muestra en la primera línea
Campus:	El campus donde hizo su último checkin el contacto. Este campo no es obligatorio y se muestra en la segunda línea
Edificio:	Edificio donde hizo el último checkin el contacto. Este campo no es obligatorio y se muestra en la segunda línea
Localización interior:	Es un string formado por el número de edificio, la planta y el nombre de la localización unidos por puntos. Este campo no es obligatorio y se muestra en la tercera línea.
Fecha:	Fecha del último checkin. Este campo no es obligatorio y se muestra en la tercera línea
Foto	Imagen usada por el contacto en su perfil de Facebook. Este campo es obligatorio y se muestra en el lateral izquierdo de la celda.

La búsqueda de contactos consiste en una tabla donde se lista en celdas los contactos que el usuario tiene en Facebook y que usan la aplicación Find My Mates. En cada celda se muestra el nombre del contacto en Facebook y la foto que usa en su perfil.

Esta pantalla permite seleccionar celdas para poder añadir los contactos mediante el botón de la barra de navegación.

El Mapa muestra un mapa de google maps centrado en la posición del usuario. En este se muestran las marcas de donde se encuentran los contactos. La información de estas marcas es la misma que se muestra en la vista principal cuando el contacto tiene hecho checkin.

3.3.2.2 Buildings

Esta pestaña consiste en un control de navegación con dos vistas. La principal en la que se muestran una tabla con celdas para cada edificio que el usuario ha añadido para hacer auto checkin y la segunda para seleccionar los edificios que se quieran añadir de la misma manera que se hace con los contactos. Las celdas se agrupan en secciones para cada campus.

Las celdas Muestran el nombre del edificio como campo obligatorio.

3.3.2.3 Checkin

Esta pestaña consiste en una única pantalla y una ventana de confirmación.

La pantalla es una tabla de celdas para cada localización cercana al usuario.

Las celdas consisten en dos líneas de texto. La primera describe la nomenclatura de la localización descrita para la pantalla de contactos. La segunda línea se muestra el nombre del edificio y el campus.

3.3.2.4 Settings

La pestaña de settings es una tabla con dos celdas. La primera contiene un interruptor para activar y desactivar el auto check-in.

La segunda contiene el nombre del usuario registrado, el botón FBLoginView del SDK de Facebook que permite al usuario registrarse o cerrar la sesión. Esta celda también contiene un FBProfilePictureView para mostrar la foto de perfil del usuario registrado.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

3.3.3 Funciones.

En este apartado se describen de manera detallada las interacciones con el sistema que resuelven cada una de las funciones.

El formato que se sigue es el de historias de usuario describiendo en un primer punto que es lo que como usuario se desea que haga la aplicación y en un siguiente paso los criterios que satisfacen y prueban que la funcionalidad se cumple.

3.3.3.1 Historias de usuario

Una historia de usuario establece la unidad mínima de funcionalidad que es de utilidad para un usuario del sistema.

Constan de un identificador, un título, una descripción y unas pruebas de aceptación que también pueden ser entendidas según bibliografía como criterios de aceptación. Además en nuestro caso quedan trazados con requisitos hacia requisitos y casos de uso.

La descripción del caso de uso sigue un patrón en el que se indica lo que quiere el usuario y el motivo de porqué lo quiere. Este motivo es de utilidad para terminar de entender el cometido del sistema y orientar el diseño.

Los criterios de aceptación son una parte fundamental de la historia de usuario ya que en ellos se especifica y la interacción que tendrá el sistema con el usuario. De gran utilidad para realizar o matizar un diseño de la interfaz de usuario y para ser usado como primera “quality gate” a la hora de determinar si la historia de usuario está lista para ser liberada.

Identificador:	US-01	Título:	Ver contactos
Descripción			
Como usuario de la aplicación FMM quiero ver un listado con los amigos añadidos al sistema para ver sus localizaciones o eliminarlos cuando dejen de tener interés sus checkins			
Secuencia de pasos			

1. Seleccionar la pestaña de contactos

Historia de usuario 1: Ver contactos

Identificador:	US-02	Título:	Añadir amigos
Descripción			
Como usuario de la aplicación FMM quiero poder seleccionar amigos que tengo en Facebook y que usen FMM para poder ver sus checkins y compartir con ellos los míos			
Secuencia de pasos			

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

- Seleccionar la pestaña de contactos
- Pulsar el botón + en la barra de navegación
- Seleccionar amigos del listado que se muestra
- Pulsar el botón Save en la barra de navegación

Historia de usuario 2: Añadir amigos

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Identificador:	US-03	Título:	Eliminar contacto
----------------	-------	---------	-------------------

Descripción

Como usuario de FMM me gustaría poder eliminar amigos añadidos previamente para no tener muchos checkins que no me resultan interesantes y dejar de compartir los míos

Secuencia de pasos

- Abrir la pestaña de contactos
- Desplazar a la izquierda la celda que se corresponde con el contacto que se quiere eliminar
- Pulsar en el botón Delete

Historia de usuario 3: Eliminar amigo

Identificador:	US-04	Título:	Ver edificios de autocheckin
----------------	-------	---------	------------------------------

Descripción

Como usuario quiero ver que edificios tengo añadidos para saber en cual está el sistema publicando de forma autónoma mi posición.

Secuencia de pasos

- Abrir la pestaña de Edificios

Historia de usuario 4: Ver edificios de autocheckin

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Identificador: US-05

Título: Añadir edificios

Descripción

Como usuario quiero elegir edificios ofrecidos por el sistema para que la aplicación FMM haga checkin automáticamente.

Secuencia de pasos

- Abrir la pestaña de edificios
- Pulsar el botón + en la barra de navegación
- Seleccionar los edificios que se desean añadir
- Pulsar el botón Save

Historia de usuario 5: Añadir edificios

Identificador: US-06

Título: Eliminar edificio

Descripción

Como usuario quiero eliminar edificios para que el sistema deje de publicar mi posición cuando llego a ellos.

Secuencia de pasos

- Abrir la pestaña de edificios
- Desplazar a la izquierda la celda que se corresponde con el edificio que se desea eliminar
- Pulsar el botón Save

Historia de usuario 6: Eliminar edificios

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Identificador: US-07**Título: Checkin manual****Descripción**

Como usuario quiero hacer checkin manual para publicar a mis amigos una localización cercana ofrecida por el sistema más precisa que la localización de edificios usada en el checkin automático.

Secuencia de pasos

- Abrir la pestaña de Check-in
- Pulsar la celda correspondiente con el lugar en el que se desea hacer check-in
- Confirmar acción en la ventana emergente

Historia de usuario 7: Checkin manual**Identificador: US-08****Título: Checkin autónomo****Descripción**

Como usuario quiero que el sistema publique automáticamente cuando llego a los edificios que he añadido a la aplicación para que mis contactos conozcan donde me encuentro.

Secuencia de pasos

- Entrar en un edificio añadido a la aplicación

Historia de usuario 8: Checkin autónomo

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Identificador: US-09

Título: Desactivar autocheckin

Descripción

Como usuario quiero poder desactivar temporalmente el seguimiento del sistema para que no se publique de forma autónoma que me encuentro en algún edificio añadido.

Secuencia de pasos

- Abrir pestaña de ajustes
- Situar el interruptor de autocheckin en la posición de apagado

Historia de usuario 9: Desactivar autocheckin

Identificador: US-10

Título: Activar autocheckin

Descripción

Como usuario quiero poder activar temporalmente el seguimiento del sistema para que se publique de forma autónoma que me encuentro en algún edificio añadido.

Secuencia de pasos

- Abrir pestaña de ajustes
- Situar el interruptor de autocheckin en la posición de encendido

Historia de usuario 10: Activar autocheckin.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Identificador: US-11	Título: Hacer login
Descripción	
Como usuario quiero poder hacer login y dar permiso a FMM para acceder a mi información básica para poder identificarme como usuario en el sistema y poder añadir amigos.	
Secuencia de pasos	
<ul style="list-style-type: none">• Abrir la pestaña de ajustes• Pulsar el botón de login de facebook	

Historia de usuario 11: Hacer login

Identificador: US-12	Título: Hacer logout
Descripción	
Como usuario quiero poder hacer logout de la aplicación para dejar de publicar mis localizaciones y poder iniciar sesión con otro usuario	
Secuencia de pasos	
<ul style="list-style-type: none">• Abrir la pestaña de ajustes• Pulsar el botón de logout de facebook	

Historia de usuario 12: Hacer logout

Identificador: US-13	Título: Ver checkins en el listado de contactos
Descripción	
Como usuario quiero poder ver las posiciones de mis contactos	
Secuencia de pasos	
<ul style="list-style-type: none">• Abrir la pestaña de contactos	

Historia de usuario 13: Ver checkins en el listado de contactos

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Identificador: US-14	Título: Ver checkins en un GIS
Descripción	
Como usuario quiero poder ver las posiciones de mis contactos en un GIS	
Secuencia de pasos	

- Abrir la pestaña de contactos
- En barra de navegación, pulsar en el botón maps.

Historia de usuario 14: Ver checkins en un GIS

3.3.4 Requisitos de rendimiento.

Identificador	Nombre
RNF-06	Networking
Descripción	
Cuando el sistema realice operaciones de networking y sea necesario mantener al usuario a la espera deberá mostrar información de proceso.	

Requisito no funcional 1: Networking

Identificador	Nombre
RNF-07	Rendimiento
Descripción	
La interfaz de usuario debe responder en todo momento al usuario. El hilo principal de la aplicación no debe quedar bloqueado por estar realizando operaciones pesadas.	

Requisito no funcional 2: Rendimiento

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

3.3.5 Restricciones de diseño.

Identificador	Nombre
RNF-08	Dispositivo
Descripción	
El sistema se desarrolla para iPhone 5s	

Requisito no funcional 3: Dispositivo

Identificador	Nombre
RNF-10	Idioma
Descripción	
El sistema muestra el contenido en inglés	

Requisito no funcional 4: Idioma

3.3.6 Atributos del sistema.

La aplicación debe ser fiable. Para ello deberá tener en cuenta algunas excepciones al funcionamiento como no tener conectividad a internet o no estar registrado. En estos casos debe mostrarse una alerta al usuario para que este entienda el erróneo funcionamiento de la aplicación.

No se contemplan requerimientos de portabilidad para este proyecto. Es por tanto una aplicación desarrollada para ser ejecutada de manera fiable en un dispositivo iPhone 5s que ejecute la versión 8.1 del sistema operativo.

La seguridad queda delegada en los servicios externos que usa la aplicación. No se contemplan requerimientos de seguridad en el servicio web que almacena los checkins

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Capítulo 4: Diseño

Contenido

- 4 DISEÑO50
 - 4.1 ALTERNATIVAS DE DISEÑO.....50
 - 4.2 DISEÑO51
 - 4.2.1 Find My Mates Web Service.....51
 - 4.2.2 Find My Mates App.....59

4 Diseño

En este capítulo se detalla cómo se ha diseñado el sistema. En un primer punto se ven dos diseños alternativos. Se parte de un diseño básico que es evolucionado y finalmente implementado.

4.1 Alternativas de diseño

Al tener el requisito no funcional de tener un almacén de datos centralizado al cual poder acceder todos los dispositivos que ejecutan Find My Mates se barajan dos opciones para cubrir este requisito.

El motivo por el que se necesita este almacén centralizado es para tener un lugar donde almacenar los checkins de los usuarios y que sea accesible por los contactos del usuario que hace un checkin.

Como primera alternativa se muestra a continuación un gráfico en el que se representa como el terminal móvil se conecta directamente al almacén de datos. El código de la aplicación debe implementar las consultas remotas y la lógica de negocio que le permite saber quién es amigo y quien no para mostrar los checkins correspondientes.



Alternativa de diseño 1

Este diseño obligaría a re implementar la lógica de acceso a los checkins si se desarrollase una aplicación para Android o Windows mobile teniendo en este caso implementaciones distintas para cubrir un solo requisito. Esta razón es suficiente como para mejorar el diseño trasladando esta lógica a un servidor de aplicaciones y exponerla a través de un servicio web como se muestra a continuación.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Alternativa de diseño 2

Con este diseño la aplicación se abstrae de cómo se obtienen los checkins y lo único que tiene que hacer es preguntar a la API del servicio web y es el diseño escogido para llevar a cabo la ejecución del proyecto.

4.2 Diseño

En este capítulo se detalla el diseño de la solución propuesta para Find My Mates. Consta de dos componentes de software diferenciados. Uno se ejecuta en Google App Engine y otro que se ejecuta en un iPhone conformando así un sistema cliente-servidor.

El primero es el servidor y ya que se trata de un servicio web, está denominado como Find My Mates Web Service, o FMMWebService.

El segundo es el cliente y, al tratarse de la aplicación para el teléfono, está denominado Find My Mates App o FMMApp y es dependiente del primero.



Para el desarrollo de este documento se detallan por separado cada uno de los componentes.

4.2.1 Find My Mates Web Service

Es un servicio web desarrollado en Python 2.7 que se ejecuta en Google App Engine. En la arquitectura cliente-servidor hace el papel de servidor, es por tanto un componente pasivo que responde a las peticiones de un cliente a través de la API REST que implementa.

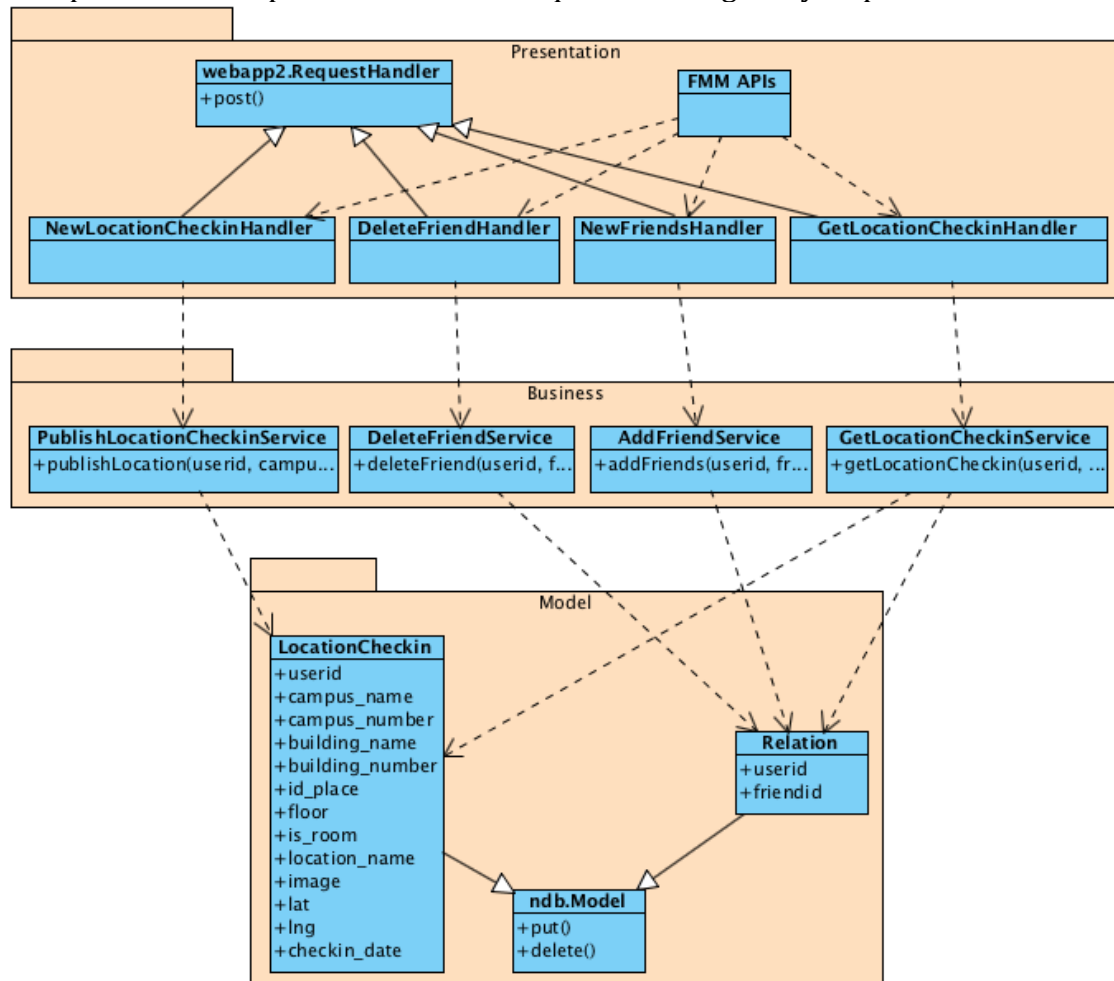
Su construcción está basada en tres capas: presentación, negocio y datos.

La capa de presentación expone la API REST y contiene todas las clases especializadas en la captura de las llamadas a cada servicio de la API y ofrecer la respuesta al cliente.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

La capa de negocio implementa todas las reglas de negocio para cada uno de los servicios ofrecidos implementando cada caso de uso del api en una clase.

La capa de datos implementa el modelo que usa el negocio y su persistencia.



4.2.1.1 FMM APIs

Función	Método HTTP
getlocationcheckin	POST
Descripción	
Esta función devuelve el último checkin del amigo que se solicita por parámetros. Este checkin es la serialización en JSON de un objeto LocationCheckin.	
Parámetros	Descripción
userid	El identificador del usuario que hace la petición
friendid	El identificador del usuario del cual se quiere conocer su última localización publicada

API 1: getlocationcheckin

Función	Método HTTP
newlocationcheckin	POST
Descripción	
Esta función permite publicar un checkin a los amigos del usuario que hace la petición	
Parámetros	Descripción

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

userid	El identificador del usuario que hace la petición
campus_name	El campus donde el usuario publica que se encuentra
campus_number	El número correspondiente al campus donde se encuentra el usuario.
building_name	Se corresponde con el nombre del edificio donde se encuentra el usuario.
building_number	El número que se corresponde con el edificio donde se encuentra el usuario
id_place	Es el identificador del emplazamiento concreto en el que se encuentra el usuario
floor	Piso en el que se encuentra el emplazamiento donde el usuario hace el checkin
location_name	Nombre del emplazamiento concreto en el que se encuentra el usuario
is_room	Enumerado que indica el tipo del emplazamiento
image	URL de la imagen que se corresponde con el checkin
lat	La latitud en la que se encuentra el lugar del checkin
lng	Longitud geográfica en la que se encuentra el lugar del checkin

API 2: newlocationcheckin

Función	Método HTTP
addfriendslist	POST
Descripción	
Esta función permite añadir una lista de amigos. La lista es un array en JSON con los identificadores de los amigos que se desea añadir.	
Parámetros	Descripción
userid	El identificador del usuario que hace la petición
friendsdids	Lista en JSON con los identificadores de los amigos que el usuario desea añadir

API 3: addfriendslist

Función	Método HTTP
deletefriend	POST
Descripción	
Esta función permite eliminar un amigo previamente añadido	
Parámetros	Descripción
userid	El identificador del usuario que hace la petición
friendid	Identificador del amigo que se desea eliminar

API 4: deletefriend

4.2.1.2 Business Layer

PublishLocationCheckinService	
Esta clase implementa las reglas de negocio para el caso de uso de hacer checkin en el servidor independientemente de que sea manual o automático.	
Método	Descripción

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

publishLocation	Publica la localización del usuario recibiendo los parámetros que se corresponden con los atributos de la clase del modelo LocationCheckin a excepción de la fecha y hora que lo autogenera la base de datos.
Clase 1: PublishLocationCheckinService	

GetLocationCheckinService	
Esta clase implementa las reglas de negocio para el caso de uso en el que el usuario del sistema ve las localizaciones de sus amigos. Esta clase es responsable de devolver solamente las localizaciones de los usuarios que tienen al usuario solicitante como amigo.	
Método	Descripción
getLocationCheckin	Devuelve la localización del usuario solicitado en el parámetro friendid si este es amigo el usuario que hace la petición. userid es el parámetro que lleva el identificador del usuario que hace la petición.
Clase 2: GetLocationCheckinService	

AddFriendService	
Esta clase implementa las reglas de negocio para el caso de uso en el que el usuario del sistema desea añadir amigos.	
Método	Descripción
addFriends	Añade al sistema los identificadores de los usuarios pasados en una lista json en el parámetro friendsids. userid es el parámetro que lleva el identificador del usuario que hace la petición.
Clase 3: AddFriendService	

DeleteFriendService	
Esta clase implementa las reglas de negocio para el caso de uso en el que el usuario del sistema desea borrar un amigo.	
Método	Descripción
deleteFriend	Elimina del sistema el identificador del usuario pasado en el parámetro friendid. userid es el parámetro que lleva el identificador del usuario que hace la petición.
Clase 4: DeleteFriendService	

4.2.1.3 Data Layer

La capa de datos contiene las clases de los objetos del dominio. Estas son clases manejadas que conocen el sistema de persistencia y son capaces de almacenarse mediante el método put() y borrarse mediante el método delete().

LocationCheckin	
Los objetos de esta clase se son responsables de almacenar las entidades del dominio correspondientes a los checkins de los usuarios.	
Atributos	Descripción
userid	El identificador del usuario que hace el checkin
campus_name	El campus donde el usuario publica que se encuentra

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

campus_number	El número correspondiente al campus donde se encuentra el usuario.
building_name	Se corresponde con el nombre del edificio donde se encuentra el usuario.
building_number	El número que se corresponde con el edificio donde se encuentra el usuario
id_place	Es el identificador del emplazamiento concreto en el que se encuentra el usuario
floor	Piso en el que se encuentra el emplazamiento donde el usuario hace el checkin
location_name	Nombre del emplazamiento concreto en el que se encuentra el usuario
is_room	Enumerado que indica el tipo del emplazamiento
image	URL de la imagen que se corresponde con el checkin
lat	La latitud en la que se encuentra el lugar del checkin
lng	Longitud geográfica en la que se encuentra el lugar del checkin
checkin_date	Es de tipo <code>ndb.DateTimeProperty</code> en UTM. El sistema de persistencia lo genera automáticamente por cada entidad que se genera de este tipo.

Clase 5: LocationCheckin

Relation

Los objetos de esta clase se son responsables de almacenar las entidades del dominio correspondientes a las relaciones entre usuarios.

Atributos	Descripción
userid	El identificador del usuario que quiere ser amigo de friendid.
friendid	El identificador del usuario del que se quiere ser amigo.

Clase 6: Relation

4.2.1.4 Diagramas de secuencia

La primera secuencia que se muestra se corresponde con el caso de uso en el que la aplicación del móvil solicita al servidor el último checkin de alguno de sus contactos. La función de la API consultada es `getlocationcheckin` que debe devolver la localización como un objeto `LocationCheckin` serializado en un string JSON.

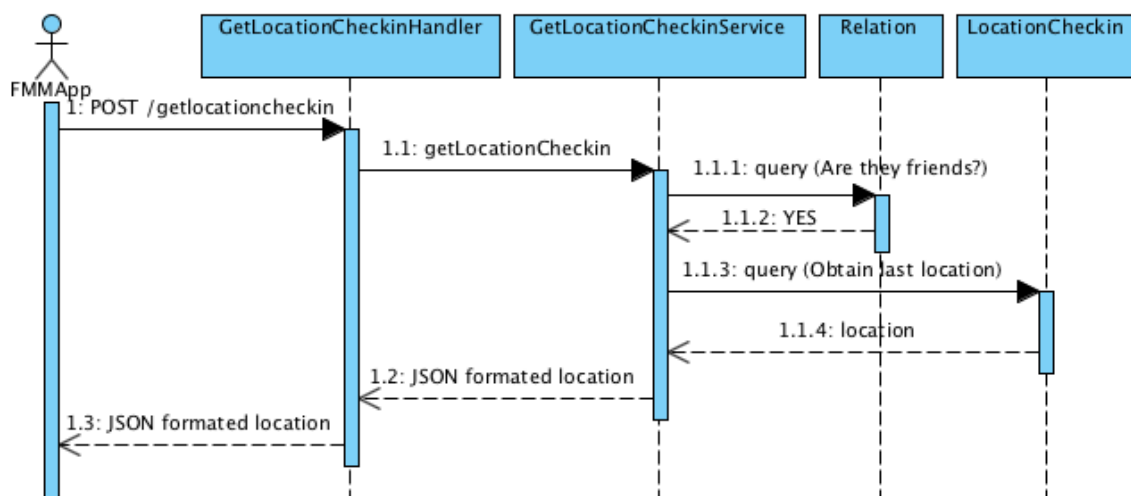


Diagrama de secuencia 1: Obtener la posición de un amigo

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

La API recoge la llamada y sus parámetros para crear la llamada al negocio usando el método getLocationCheckin.

Este método implementa una consulta a la base de datos para verificar la amistad de los usuarios y en caso afirmativo genera otra consulta para obtener el último checkin.

Este es serializado en JSON y devuelto a la capa de presentación para que sea devuelto como resultado de la petición http.

La siguiente secuencia se corresponde con la publicación de un checkin.

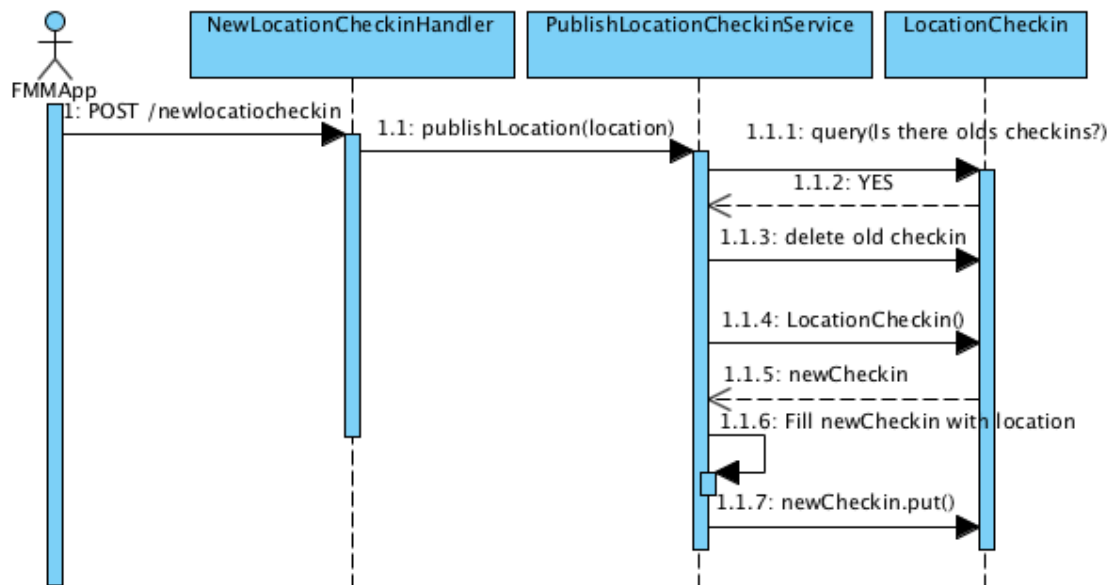


Diagrama de secuencia 2: Publicar una posición

Para publicar un checkin la aplicación debe hacer uso de la función de la API new locationcheckin.

El manejador de la llamada recoge los parámetros y emite la llamada a la capa de negocio mediante el método publishLocation.

Este método comprueba si el usuario tiene hecho checkins anteriores para borrarlos en su caso. A continuación crea el nuevo checkin a partir de los parámetros recibidos y lo almacena haciendo uso del método put() del nuevo checkin.

La siguiente secuencia se corresponde con la actividad de añadir nuevos amigos al sistema para que pase a formar parte de los contactos de la aplicación.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

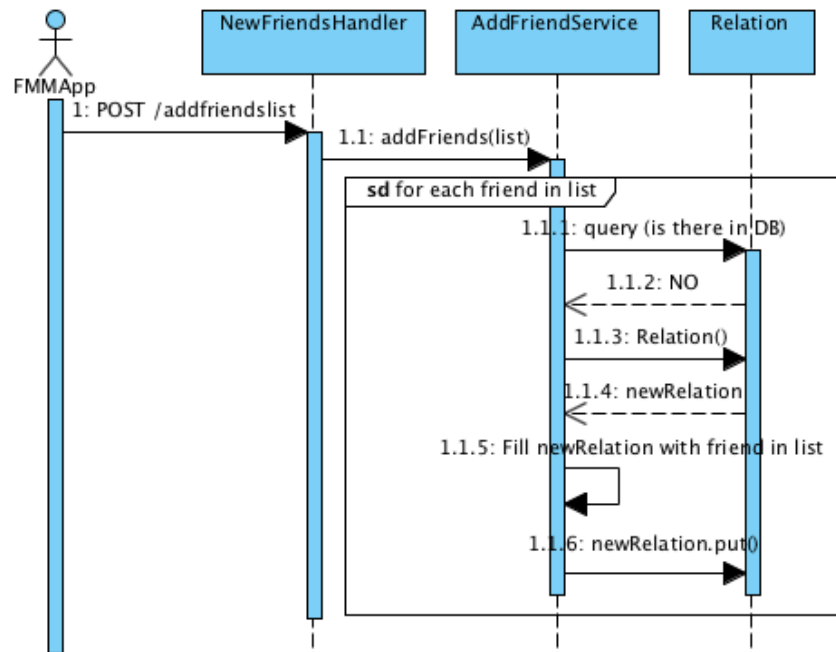


Diagrama de secuencia 3: Añadir amigo

Se inicia la secuencia cuando la aplicación en el teléfono hace la petición a la API addfriendlist. Uno de los parámetros que debe recibir la API es un listado con identificadores de los usuarios que se quieren añadir. Este listado al ser un array en JSON debe ser deserializado. El método de la capa de negocio addFriends recibe el JSON y tras la deserialización itera por cada elemento del array.

Cada elemento se corresponde con el identificador de un usuario para añadir por lo que para cada uno se verifica que no esté en la base de datos ya añadido como contacto.

Si no se encuentra añadido, se crea la nueva relación y se almacena.

La siguiente secuencia se corresponde con el borrado de un contacto.

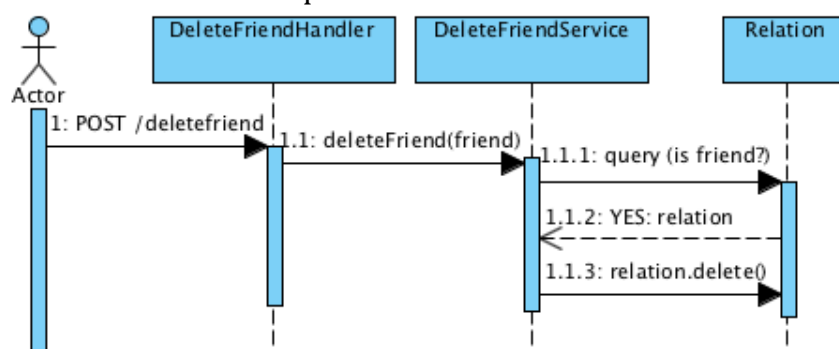


Diagrama de secuencia 4: Borrar amigo

Para este cometido la aplicación en el móvil debe usar la API deletefriend. Esta recibe por parámetros el identificador del amigo que se desea eliminar.

El manejador de la llamada usa el método deleteFriend de la capa de negocio que se asegura en primer lugar de verificar que los usuarios son amigos entre sí. Esta verificación se aprovecha para obtener el objeto de negocio que modela la relación entre los dos usuarios.

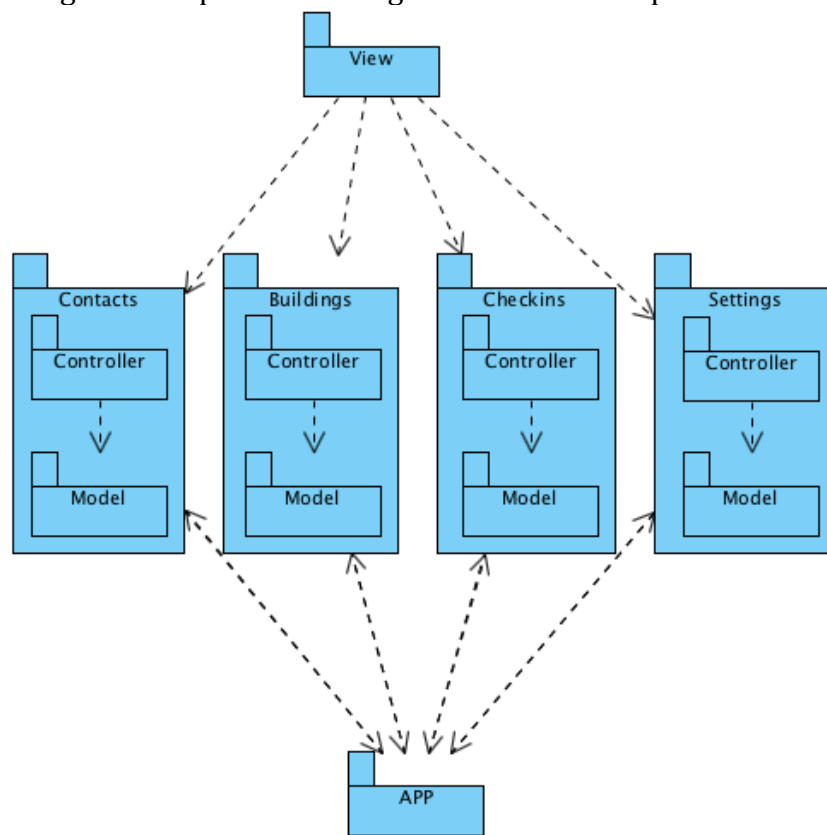
En caso afirmativo elimina la relación usando el método delete del objeto obtenido.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

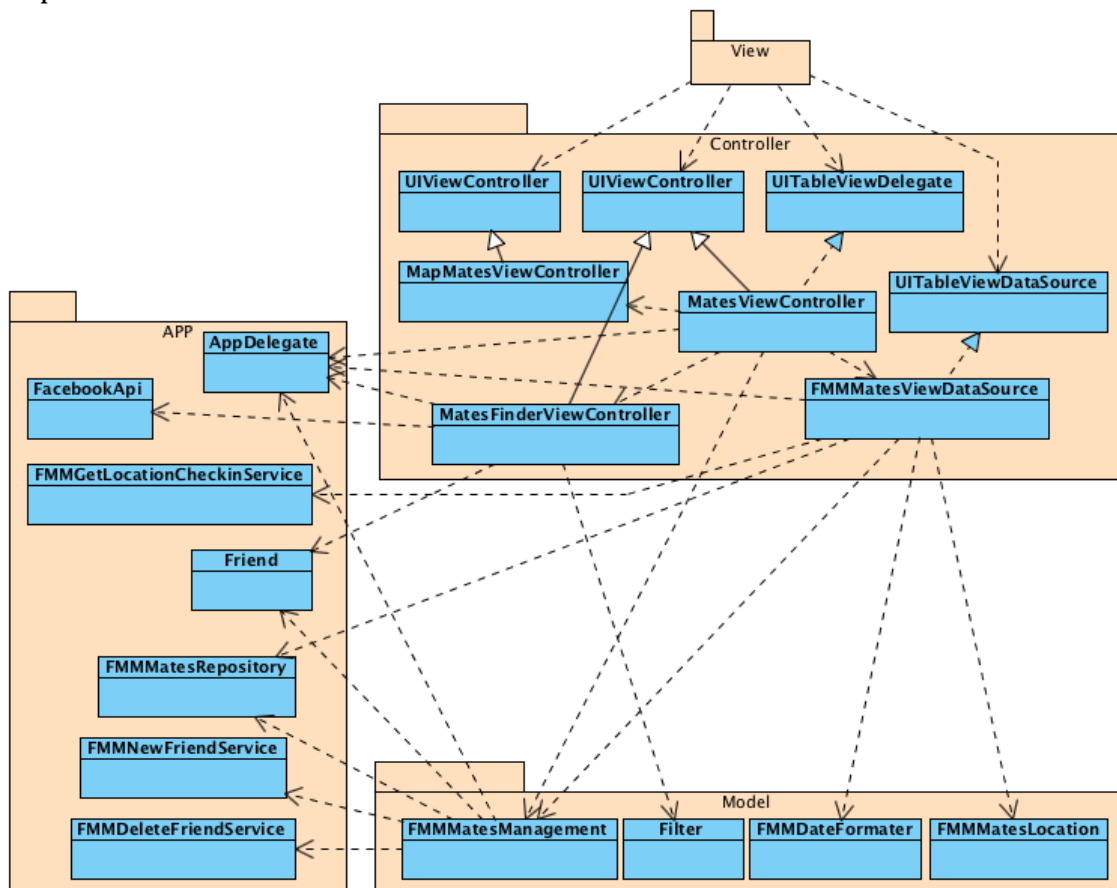
4.2.2 Find My Mates App

En este apartado se describe el diseño del componente que se despliega en el iPhone. Para tratar de hacerlo entendible y manejable, se ha dividido en cuatro módulos que tratan de corresponderse con las pestañas de las que está conformada la aplicación: contacts, buildings, checkins y settings. Cada uno de estos módulos trata de seguir una arquitectura modelo vista controlador como propone el manual de desarrollo de Apple. Aparte se ha añadido un módulo secundario llamado App que se encarga de inicializar los módulos principales y proporcionarles de ciertos servicios. El código que contiene pertenece al modelo de la aplicación pero al ser común para el resto de módulos se desarrolla en un módulo aparte.

El código de la vista en este proyecto es autogenerado a partir del story board por lo que en los siguientes apartados no figuran clases correspondientes a esta capa.



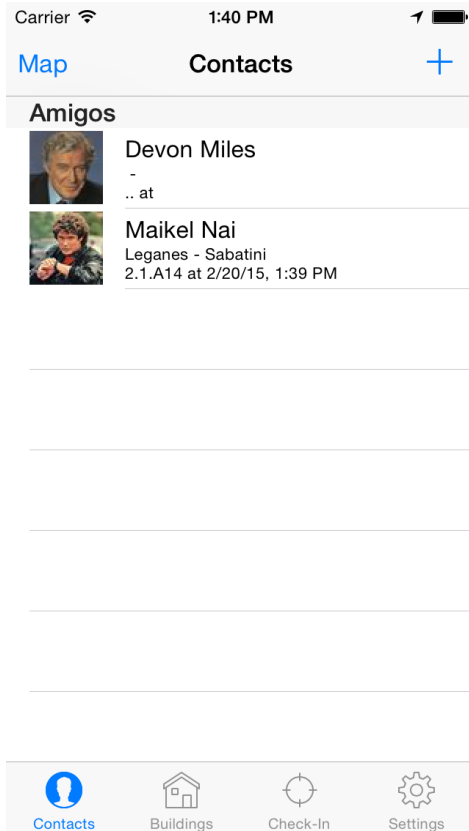
Este módulo contiene la gestión y las distintas visualizaciones de los contactos junto a su información de posicionamiento que publiquen. Contiene tres pantallas: La principal donde se listan los contactos añadidos por el usuario, la de búsqueda donde se listan los amigos que tiene el usuario en Facebook y no están en la lista de contactos y el mapa donde se ven los checkins de los contactos sobre el GIS de Google Maps.



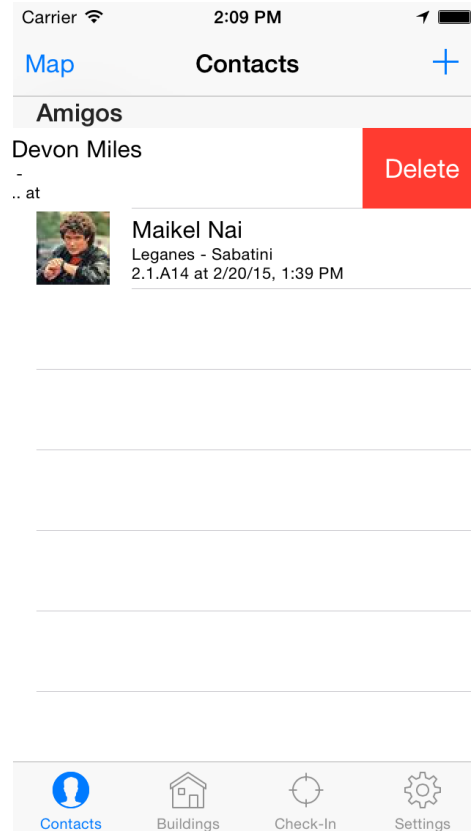
Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

4.2.2.1.1 Listado de contactos.

En esta pantalla se implementan las historias de usuario de ver contactos, eliminar amigos y ver checkins en el listado de contactos.



Pantalla 1: Listado de contactos



Pantalla 2: Listado de contactos

El controlador está dividido en dos clases: `MatesViewController` y `FMMatesViewDataSource`. Estas implementan dos protocolos que define Apple para tratar con un `UITableView`. La clase `MatesViewController` implementa el protocolo `UITableViewDelegate` y `FMMatesViewDataSource` implementa el protocolo `UITableViewDataSource`.

`UITableViewDelegate` es un protocolo para definir el comportamiento de la tabla mientras que `UITableViewDataSource` define los métodos necesarios para mostrar o editar los datos que se muestran en la tabla. Para que esto sea posible, `MatesViewController` inicializa el datasource con el repositorio de contactos, `FMMatesRepository` y se lo asigna al datasource de la tabla.

A continuación se muestra un diagrama detallado con las clases que intervienen en la lista de contactos.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

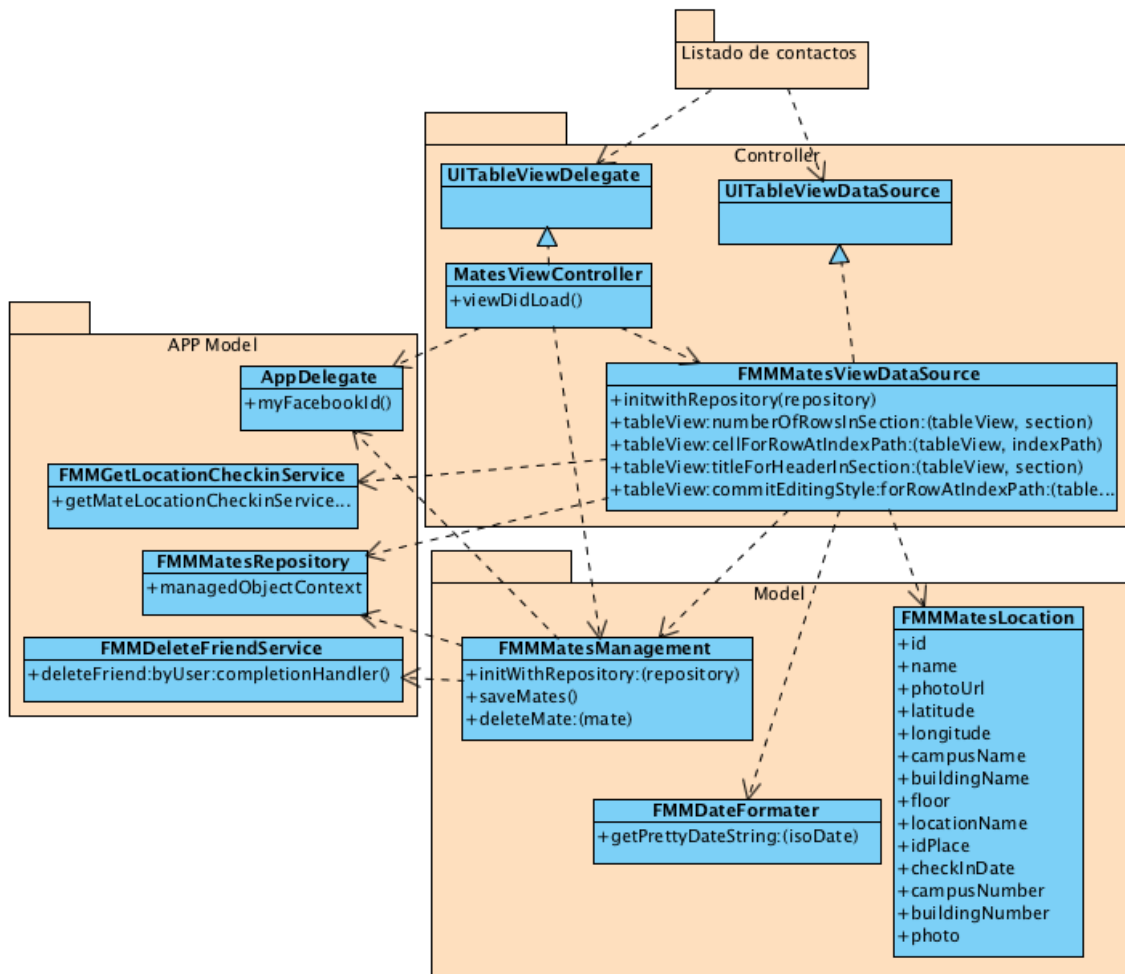


Diagrama de clases 2: Listado de contactos

Los métodos más relevantes de FMMMatesViewDataSource son el tableView:cellForRowAtIndexPath para devolver la celda con los datos para mostrar en la vista y tableView:commitEditingStyle:forRowAtIndexPath que se usa para borrar una fila y borrar el contacto.

FMMDatesLocation es una clase sin métodos que se usa para modelar el contenido de una celda de la vista.

FMMDatesFormatter contiene un método que recibe una fecha en formato iso y devuelve el string que se visualiza mostrando la fecha y la hora del checkin.

FMMDatesManagement es la clase encargada en almacenar y borrar contactos. En esta pantalla tiene el papel relevante del borrado.

FMMDatesRepository se encarga de la persistencia en la base de datos del teléfono. En este caso es responsable del borrado del contacto.

FMMDelateFriendService es responsable de borrar un amigo en el servidor.

FMMLocationCheckin es la clase responsable de conectarse al servidor y obtener los checkins de los contactos.

AppDelegate sólo se usa para obtener el identificador del usuario de la aplicación.

Para ilustrar como se implementan historias de usuario de esta pantalla se detallan a continuación la secuencia de pasos para cada uno de ellos.

En primer lugar se detallan las historias de ver contactos y ver checkins en listado de contactos. Se muestran los dos juntos ya que se aprovecha el momento en el que

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

se construye la celda de un contacto para consultar en el servidor su último checkin. Haciendo la consulta en el momento de la construcción de la celda solo se pide al servidor la información de los usuarios que se van a mostrar en la pantalla.

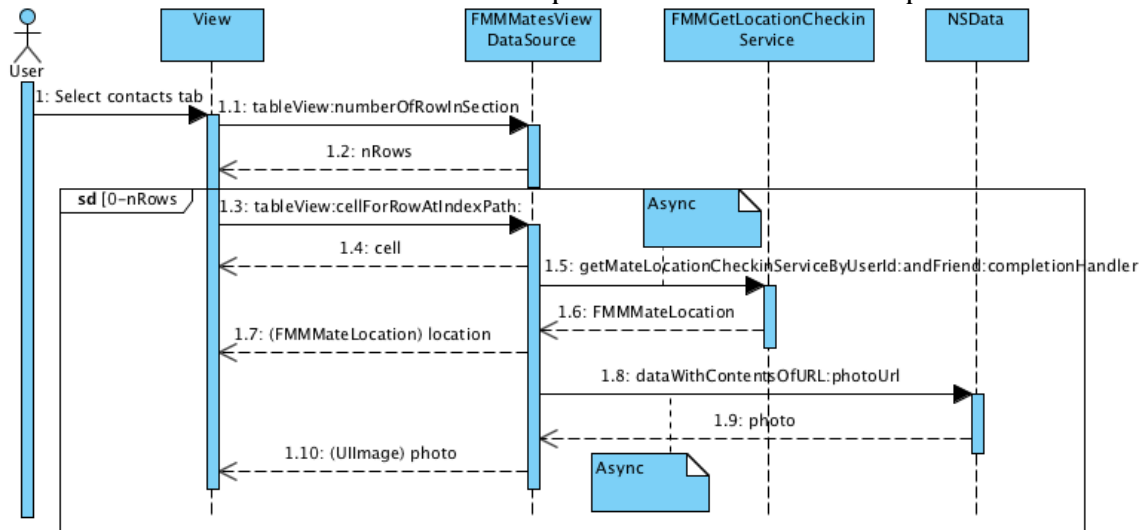


Diagrama de secuencia 5: Historia de usuario de ver contactos y ver checkins en el listado de contactos

El usuario inicia la secuencia seleccionando la pestaña de contactos. En la imagen a continuación se muestra la selección de la pestaña.



Inicio de secuencia 1: Seleccionar pestaña de contactos

Esta pestaña contiene un control UITableView que para pintarse en la pantalla necesita información que le provee el datasource. Parte de esa información es el

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

número de filas totales que el control obtiene usando el método `tableView:numberOfRowsInSection`.

En nuestro caso el número de filas se corresponde con el número de contactos añadidos al sistema ya que se mostrará la información de un contacto por fila.

A continuación, por cada celda que se va a mostrar la vista usa el método `tableView:cellForRowAtIndexPath:` para construir la celda. Esta llamada devuelve la celda con los datos de los que ya dispone el terminal, ya que falta la imagen almacenada en Facebook y el checkin almacenado en el servidor de la solución. Para alcanzar esta información se hacen dos llamadas en hilos independientes de la interfaz de usuario para no bloquearla en lo que se recuperan los datos de la red.

El `getMateLocationCheckinServiceByUserId:andFriendId:completionHandler` devuelve un objeto `FMMMateLocation` con la información devuelta por el servidor en relación al checkin del usuario identificado por el parámetro `FriendId` y el bloque referenciado en el `completionHandler` lo devuelve a la vista.

En un bloque asíncrono se obtiene la imagen de la foto y cuando se ha descargado se devuelve a la vista terminando así la ejecución de estas historias de usuario.

En el siguiente diagrama de secuencia se detallan las acciones para borrar un contacto.

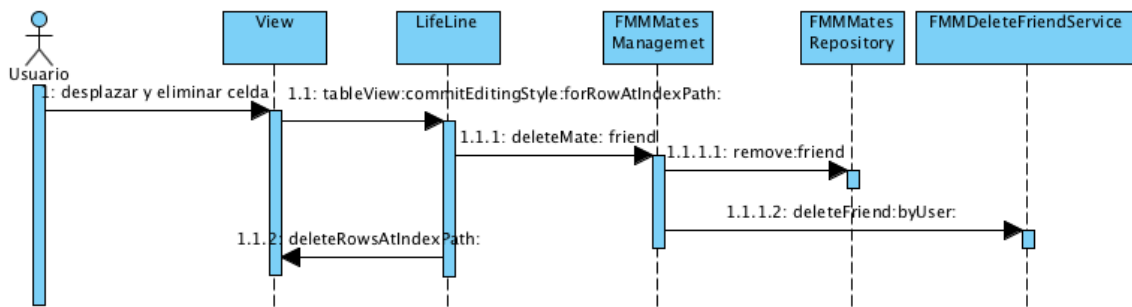
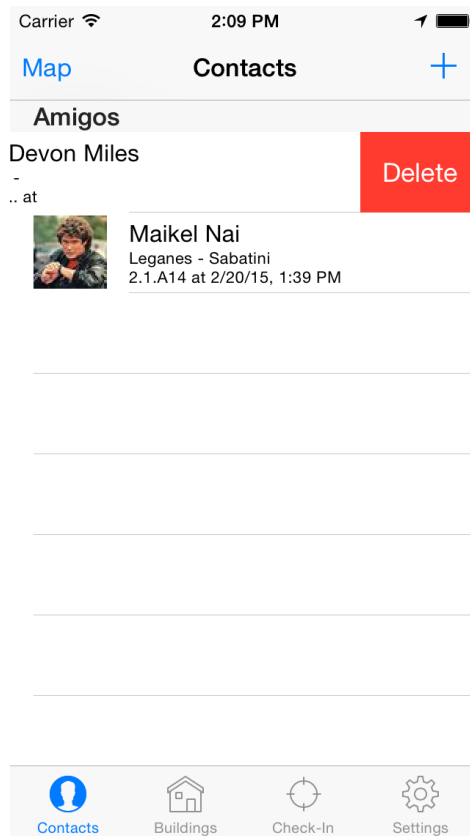


Diagrama de secuencia 6: Eliminar contacto

El usuario, mediante el gesto de desplazar una celda a la izquierda recibe la opción de eliminar la celda. Si acepta el control llama al método `tableView:commitEditingStyle:forRowAtIndexPath:`

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



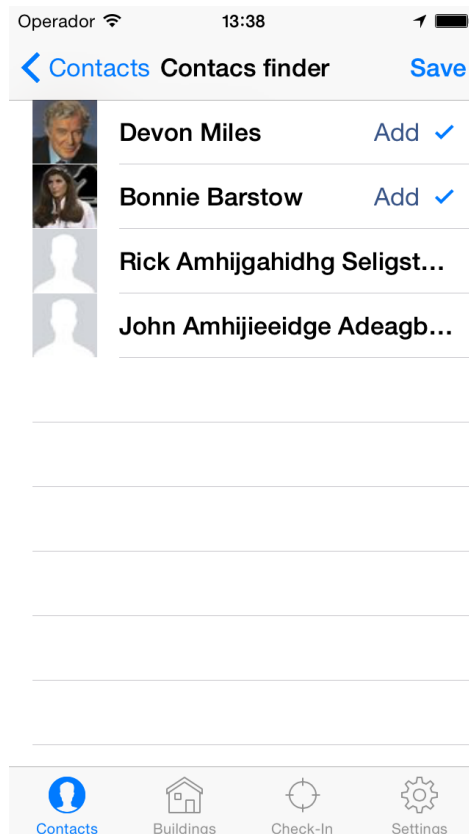
Inicio de secuencia 2: Desplazar y eliminar celda

La implementación de este método consiste en borrar la celda referenciada por el IndexPath y el usuario que se corresponde con esa celda. Esto requiere hacer dos operaciones: borrar el usuario en la base de datos usando el método del repositorio `remove:` y eliminar la relación de amistad del servidor usando el método `deleteFriend:ByUser:`

4.2.2.1.2 Búsqueda de amigos

En esta pantalla se implementa la historia de usuario de añadir amigos. Para llegar a ella el usuario debe pulsar el botón + en la pantalla de contactos.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)



Pantalla 3: Búsqueda de contactos

Para detallar la implementación se muestra en la siguiente imagen las clases que intervienen en el diseño.

Del controlador se destaca el método `getFriendList` que es usado para obtener una lista de amigos del usuario que tiene en facebook y que usan la aplicación Find My Mates.

El método `save` almacena los amigos seleccionados de la pantalla como contactos de la aplicación.

La clase `Filter` contiene un método para filtrar el listado de amigos de Facebook para no mostrar al usuario los amigos que ya tiene añadidos como contactos.

`AppDelegate` se usa para obtener la sesión de Facebook y `FacebookApi` es la responsable de conectarse a Facebook para obtener el listado de los amigos.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

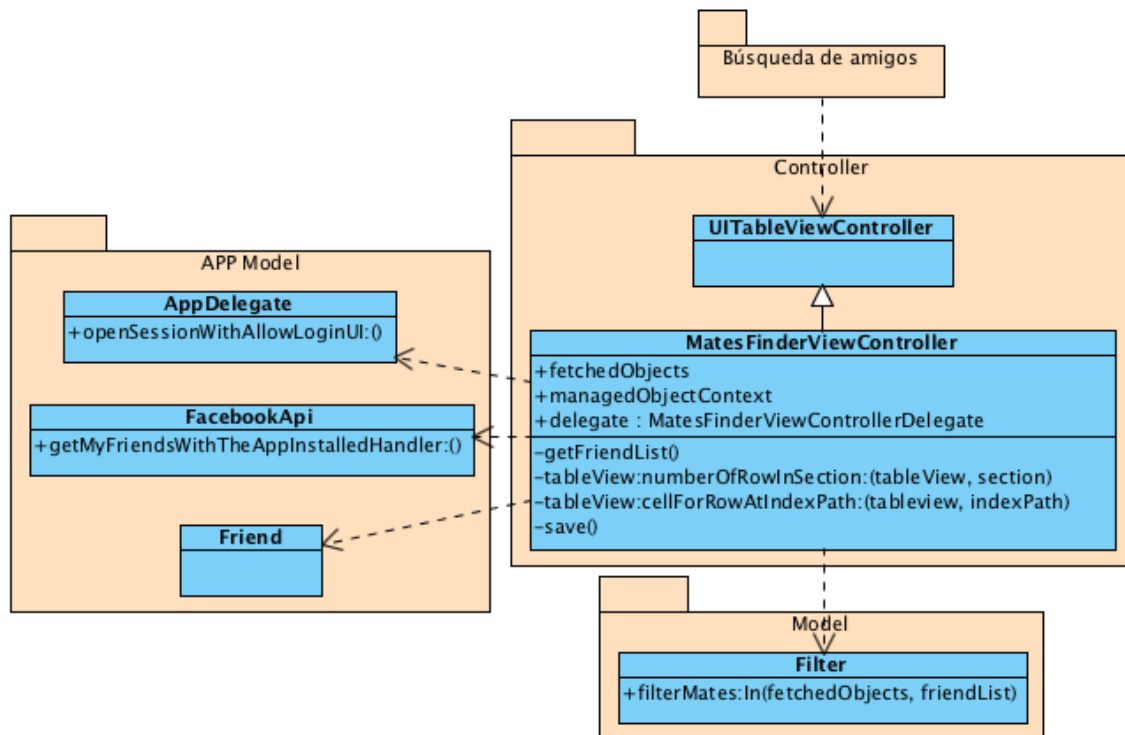
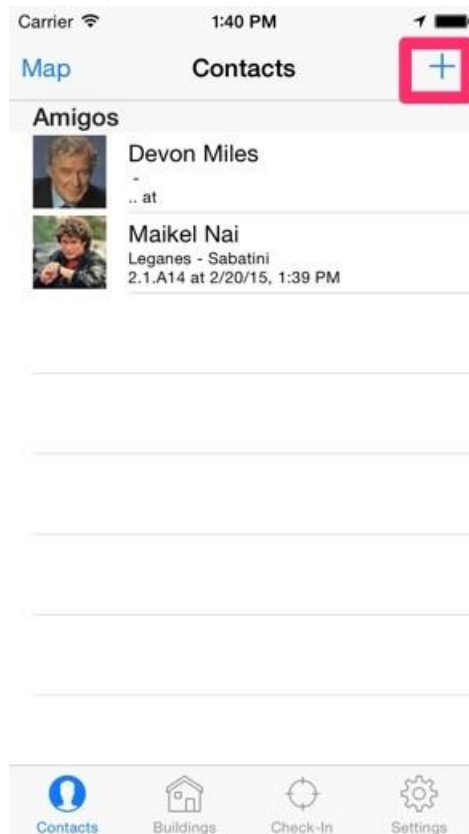


Diagrama de clases 3: Búsqueda de amigos

A continuación se muestra la secuencia seguida por el usuario y estas clases para añadir contactos.

La secuencia se inicia cuando el usuario desea acceder a la ventana de añadir contactos. Cuando esta va a aparecer el control llama al método `viewWillAppear` que es el que se usa en este diseño para obtener el listado de los amigos que el usuario tiene en Facebook mediante el método `getMyFriendsWithTheAppInstalled` de la clase `FacebookApi`.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Inicio de secuencia 3: Tap +

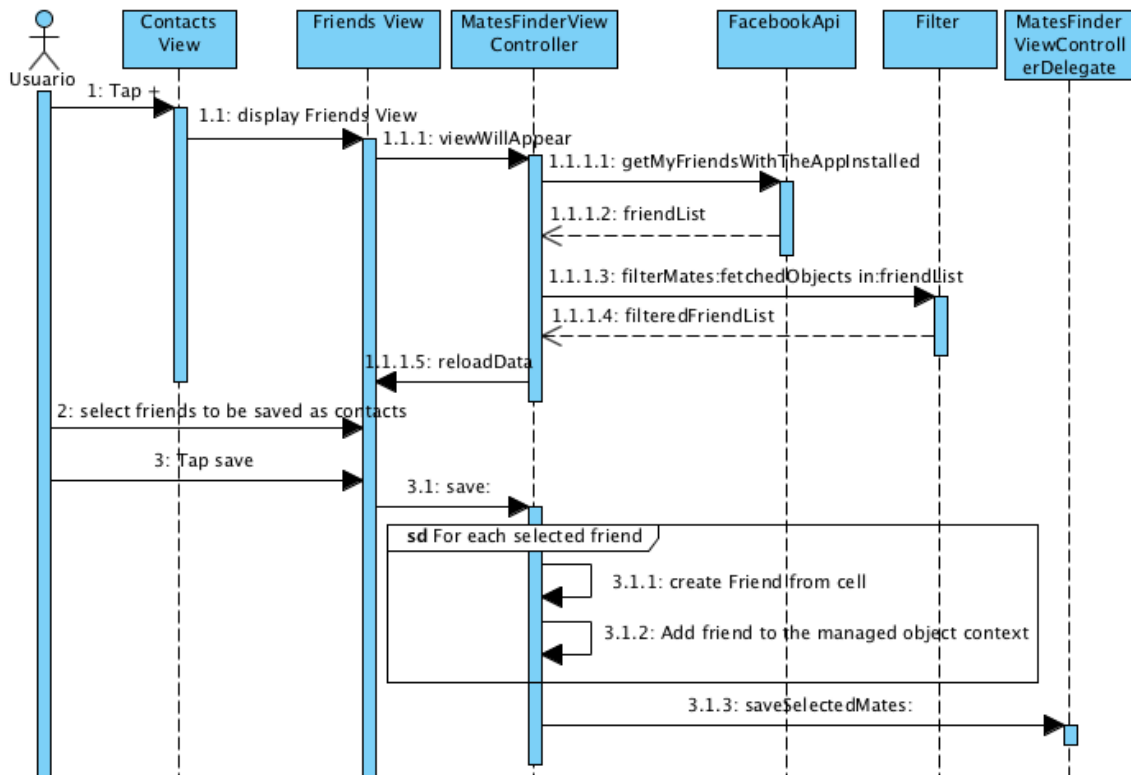


Diagrama de secuencia 7: Añadir amigos

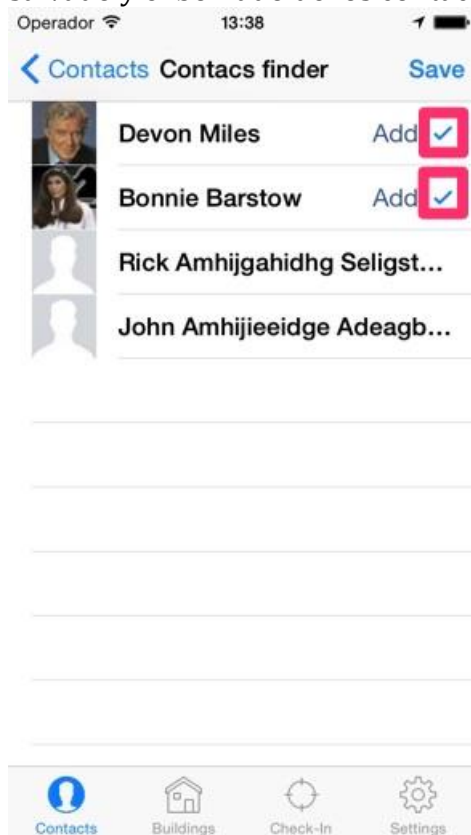
Cuando se recibe la lista se usa la clase Filter para eliminar los que ya se tienen almacenados en la base de datos de la aplicación como contactos y así mostrar en la

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

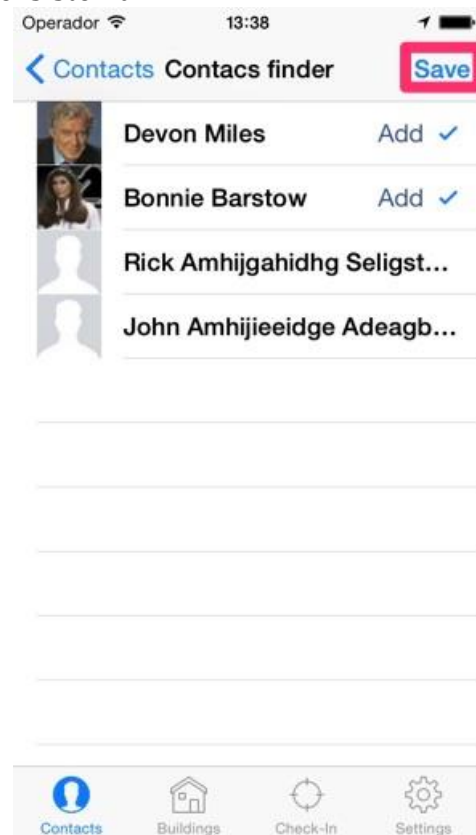
tabla sólo los que falten por añadir. Al terminar esta operación se llama al método reloadData del UITableView que tiene referenciado este controlador para desencadenar las llamadas necesarias para pintar la tabla en la vista. Estas son, como para todos los controllers de tipo UITableView, las requeridas en el protocolo UITableViewDataSource.

Una vez está la tabla rellena con los amigos usuario. Este está en disposición de seleccionarlos para después almacenarlos. El control permite selección múltiple para que el usuario pueda seleccionar varios amigos.

Una vez el usuario a terminado la selección de amigos y ejecuta la función de guardado la aplicación retorna a la ventana principal de contactos. La acción de guardado queda delegada en el controlador de la pantalla a la que se retorna ya que es este el que mantiene una referencia a la clase FMMMatesManagement, encargada del salvado y el borrado de los contactos en el sistema.



Inicio de secuencia 4: Select friend to be saved as contact

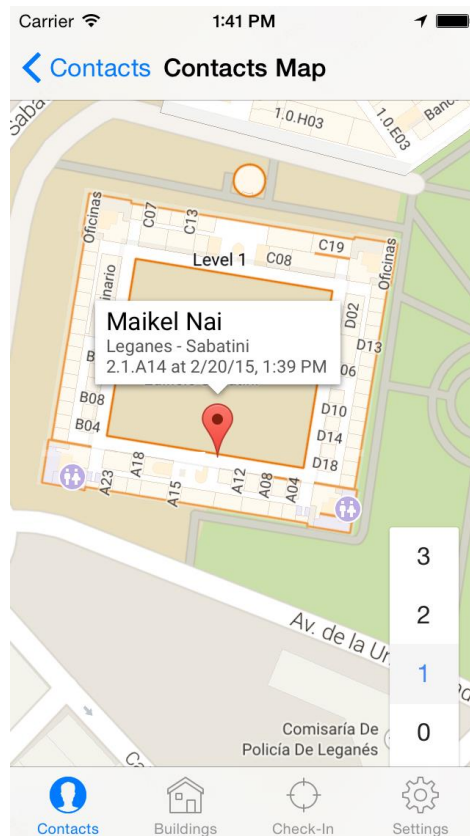


Inicio de secuencia 5: Tap save

4.2.2.1.3 Mapa

Esta vista muestra en el GIS de Google Maps los lugares donde los usuarios han hecho su último checkin. Con esto se cubre la historia de usuario de ver checkins en un GIS

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Pantalla 4: Mapa visualizando checkins de los contactos

Es la parte más sencilla del módulo de contactos. Carece de modelo ya que el `MatesViewController` proporciona al `MapMatesViewController` el modelo de lo que se debe pintar en el momento de visualizar el mapa. También se añade el controller como observador de la lectura del GPS para mover la cámara cuando el GPS indique un cambio de posición.

`viewDidLoad` inicializa el controller situando la cámara y pidiendo al sistema la posición que da actualmente el GPS.

`viewDidAppear` coloca sobre el gis las posiciones publicadas por los contactos.

`observeValueForKeyPath:ofObject:change:context` es el método que se ejecuta cuando se observa un cambio de valor en la lectura del GPS. En este método se actualiza la posición de la cámara con la posición que indica el GPS del terminal.

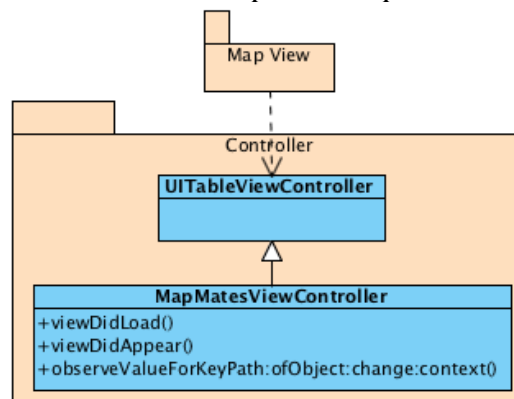


Diagrama de clases 4: Ver checkins en un Gis

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

4.2.2.2 Buildings

Este módulo permite al usuario ver, añadir y eliminar los edificios para que el sistema haga checkin de forma autónoma cuando el usuario llegue a ellos.

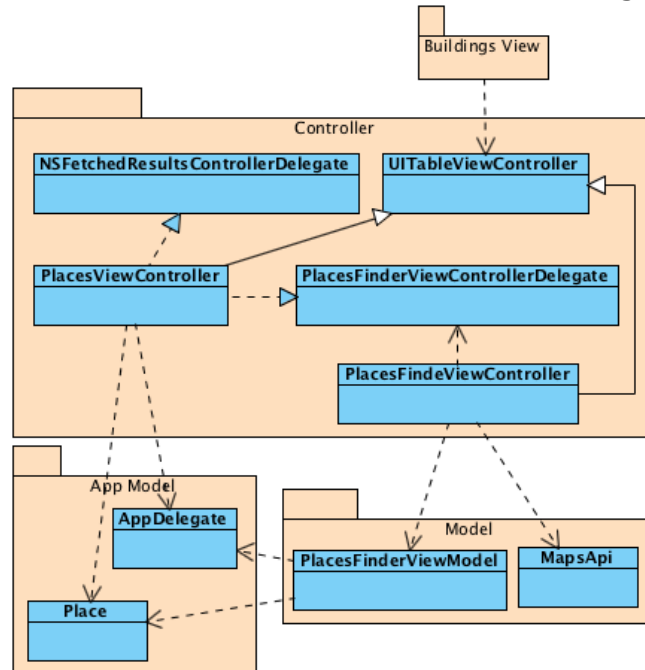


Diagrama de clases 5: Pestaña de edificios

La información de los edificios incluye sus referencias geográficas y un radio a partir del cual se puede crear una frontera geográfica para que el sistema la detecte cuando se cruza y haga checkin.

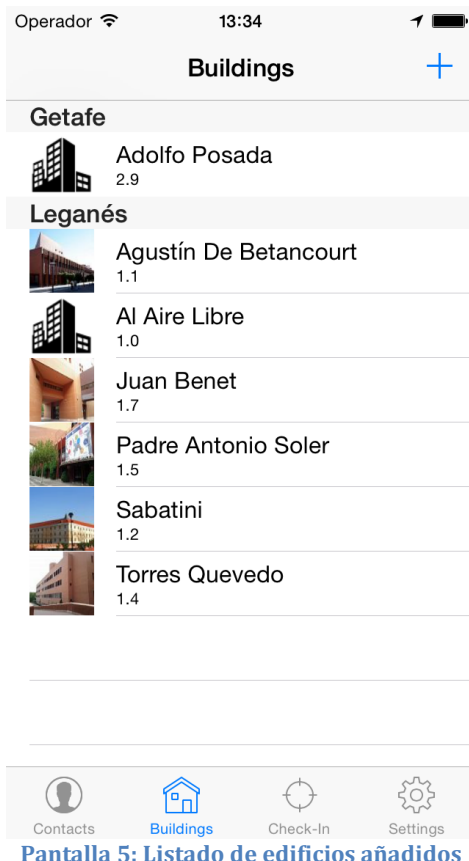
El módulo consta de dos pantallas: La principal para ver un listado de edificios añadidos que también se pueden borrar y la pantalla de búsqueda para añadir.

En la siguiente imagen se ve como se relacionan las clases que intervienen en la construcción de este módulo y en los siguientes apartados se detalla el funcionamiento por cada pantalla.

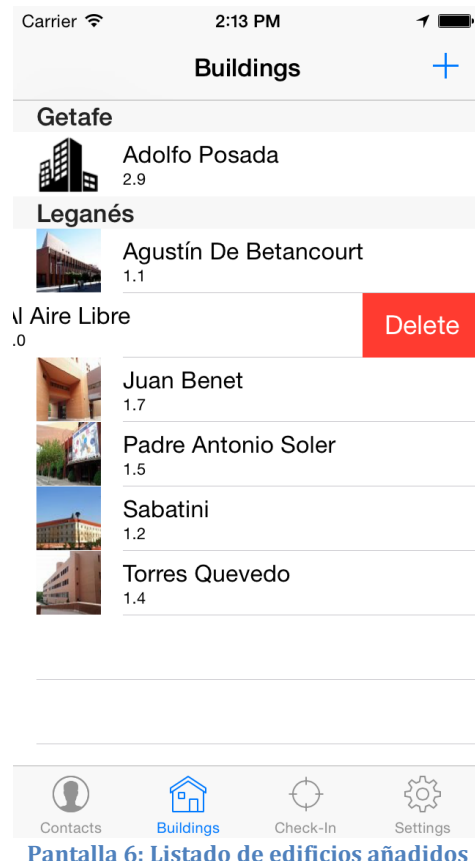
4.2.2.2.1 Listado de edificios

En esta pantalla se implementan las historias de usuario de Ver edificios de autocheckin y Eliminar edificios.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Pantalla 5: Listado de edificios añadidos



Pantalla 6: Listado de edificios añadidos

El controlador de esta vista atiende eventos de la base de datos mediante la implementación del protocolo `NSFetchedResultsControllerDelegate`. Los eventos son `controller:didChangeObject:atIndexPath:forChangeType:newIndexPath:` para cuando una entidad es añadida, eliminada o editada y para cuando una sección es cambiada, añadida, movida o borrada, el `NSFetchedResultsController` llama a `controller:didChangeSection:atIndex:forChangeType`. Antes de hacer cambios en la base de datos el `NSFetchedResultsController` llama a `controllerWillChangeContent:` y cuando termina llama a `controllerDidChangeContent`.

El método `returnControlToDelegate` lo usa el controlador de la ventana de búsqueda para indicar que se han terminado de añadir edificios. Entonces se procede a monitorizar las geo-fronteras de los edificios añadidos para notificar cuando se cruzan.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

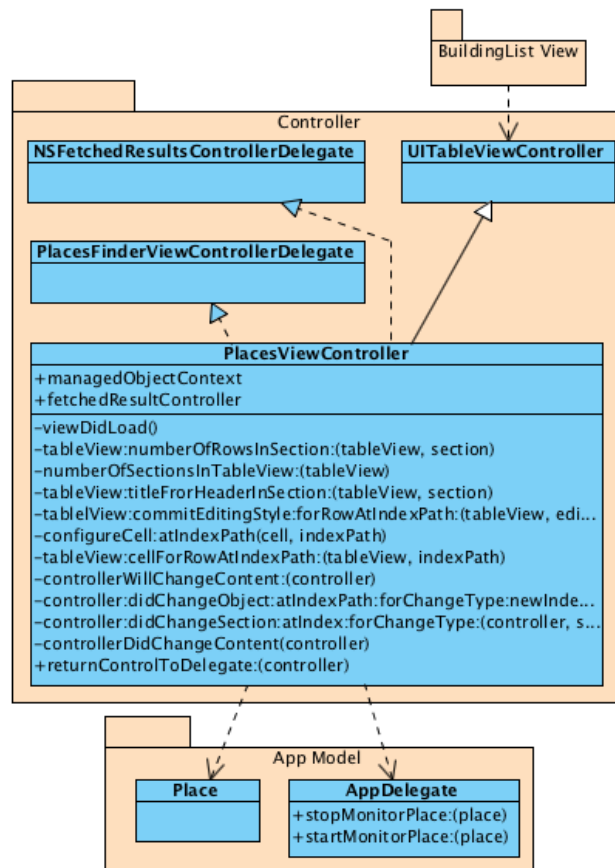


Diagrama de clases 6: Listado de edificios

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

A continuación se muestran las secuencias de ejecución para las historias de usuario de Ver edificios de autocheckin y de eliminar edificios.

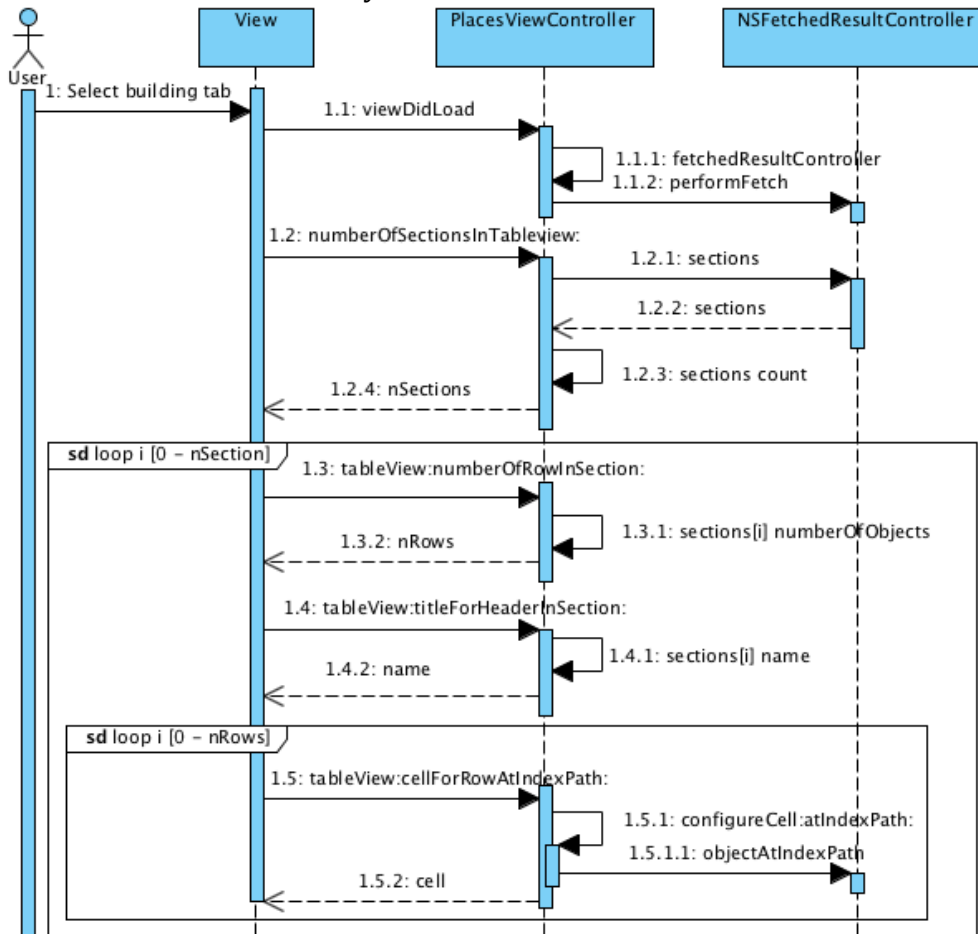
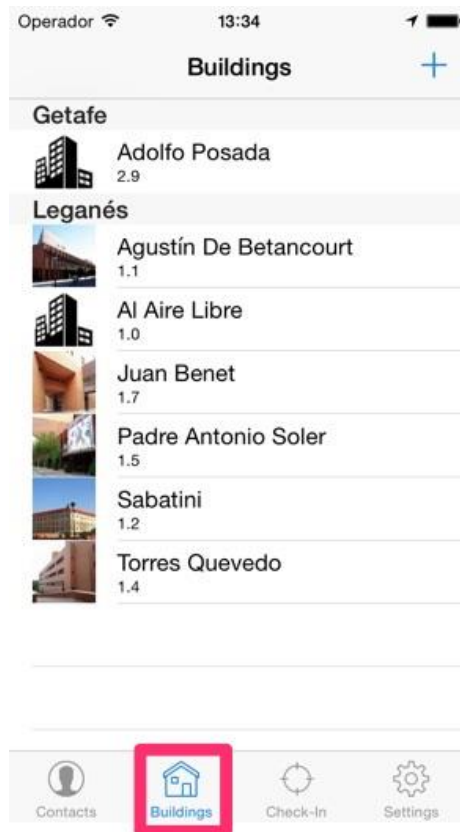


Diagrama de secuencia 8: Ver edificios de autocheckin

El usuario inicia la secuencia abriendo la pestaña de edificios. Si el control no está inicializado se ejecuta el método `viewDidLoad` para inicializar el `fetchedResultsController` y ejecutar la consulta a la base de datos que devuelve todos los edificios previamente añadidos al sistema.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Inicio de secuencia 6: Select building tab

Para que el control pueda pintarse de manera correcta necesita conocer el número de secciones que va a tener la tabla. En esta pantalla se agrupan los edificios por campus y para ello se definen las secciones por campus de manera que cuando el control llama a `numberOfSectionsInTableView`: se consultan las secciones en la base de datos para devolver el número al control.

Una vez el control conoce el número de secciones, puede empezar a rellenarlas iterando por cada una de ellas.

Se usa el método `tableView:titleForHeaderInSection`: para obtener el título de la sección que se está rellenando y el método `tableView:numberOfRowsInSection`: para obtener el número de filas que se van a rellenar en la sección.

A continuación se itera por cada celda que va a ser mostrada configurando la celda o fila en función del objeto obtenido en la base de datos para la fila en el índice pasado en el parámetro `indexPath`.

A continuación se muestra la secuencia para borrar un edificio.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

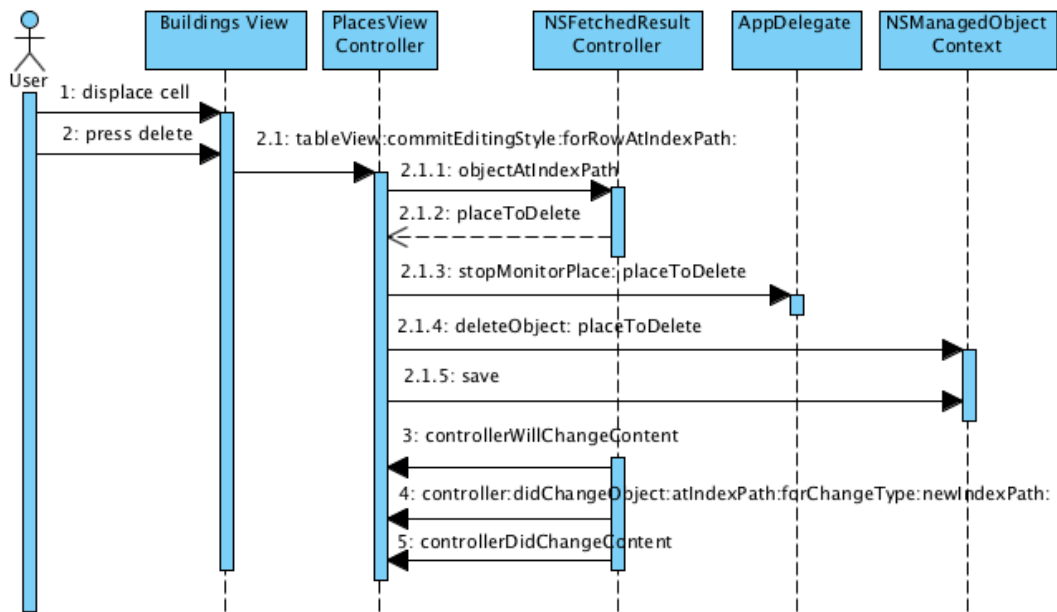
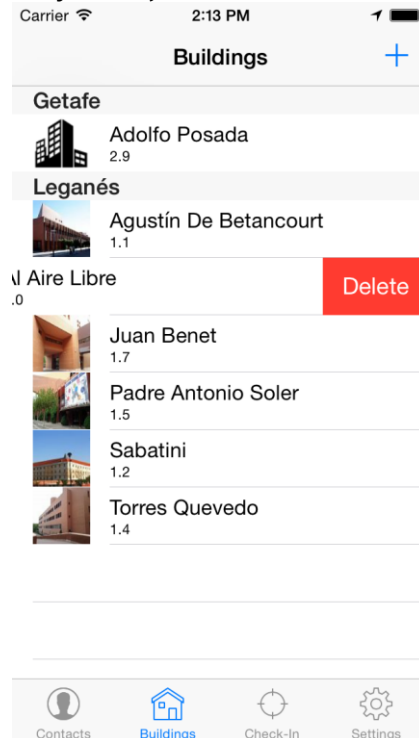


Diagrama de secuencia 9: Eliminar edificio

El usuario inicia la secuencia en la pantalla donde se muestran los edificios de autocheckin. Este, con la intención de dejar de publicar su presencia cuando llega a un edificio, desplaza a la izquierda la celda correspondiente con el edificio que desea eliminar para que aparezca el botón que confirma el borrado. Al confirmar se borra el edificio de la base de datos y se deja de monitorizar la geo-frontera del mismo.



Inicio de secuencia 7: Displace cell & press delete

Para esto, en el método 2.1, se pide al fetchedResulController que devuelva el objeto que se corresponde con el índice de la tabla que ha seleccionado el usuario para ejecutar el método del AppDelegate que detiene la monitorización de geo-fronteras.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Una vez detenida la monitorización se elimina el objeto de la base de datos. Lo cual genera unos eventos en el fetchedResulController que atiende el PlacesViewController por ser su delegado.

Primero se atiende controllerWillChangeContent que se usa para iniciar la transacción de cambios en el tableView.

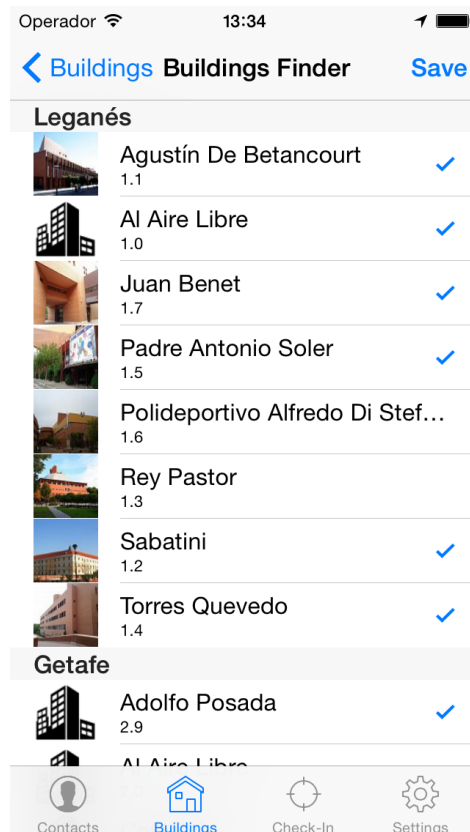
La mayoría de las veces el controlador de la vista atenderá a continuación el método 4 solamente para atender cambios en las celdas. En nuestro caso, si el usuario borra la única celda de una sección, el fetchedResulController también llamaría a controller:DidChangeSection:atIndex:forChangeType:

Cuando el fetchedResultController no tiene más cambios que notificar llama a controllerDidChangeContent: para finalizar la transacción de cambios en el tableView.

Es necesario hacer los cambios sobre el tableView dentro de una transacción para evitar inconsistencias ya que pueden sucederse varios cambios y de distinto tipo que hasta que no terminen serán incoherentes con el número de filas en secciones.

4.2.2.2.2 Búsqueda de edificios

Cuando el usuario pulsa el botón + de la barra de navegación en la pantalla de edificios accede a la pantalla de búsqueda de edificios. Está implementa la historia de usuario de Añadir edificios.



Pantalla 7: Añadir edificios

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

A continuación se detalla la implementación comenzando por el diagrama de clases que intervienen. Las clases destacables son las del modelo: PlacesFinderViewModel y MapsApi.

PlacesFinderViewModel no pretende ser el ViewModel que propone Microsoft en la arquitectura MVVM aunque en un inicio pretendía aprovechar alguna característica. Esta es una clase que abstrae a quien la usa de la base de datos de edificios. En futuras versiones deberá cambiar su nombre.

MapsApi es una clase que interroga el repositorio de mapas de la UC3M para obtener los edificios clasificados por campus.

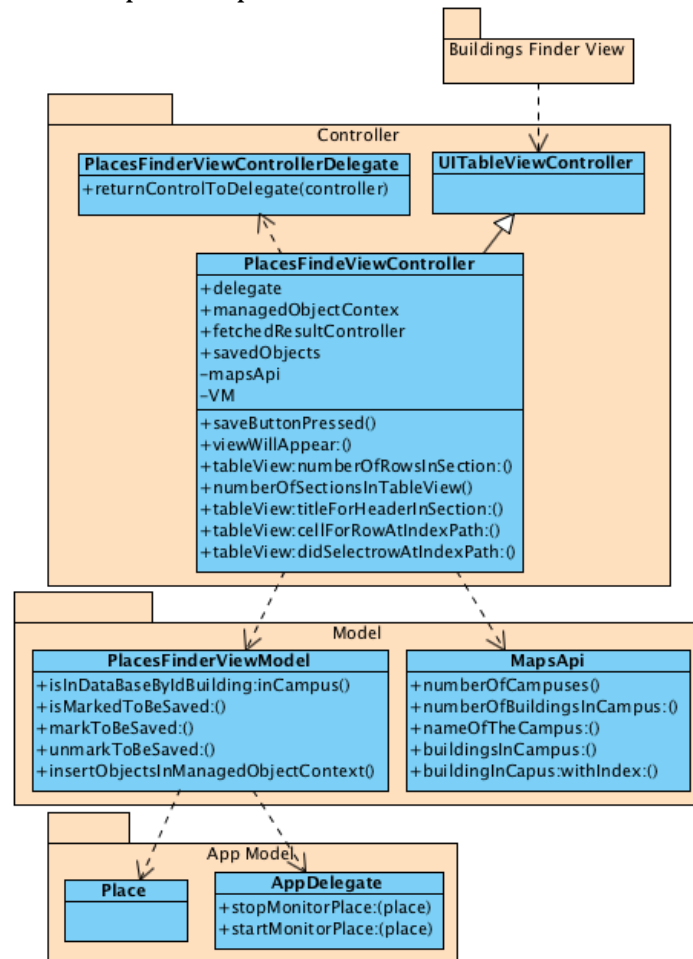


Diagrama de clases 7: Búsqueda de edificios

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

A continuación se muestra la secuencia de eventos para mostrar al usuario los edificios que se pueden añadir y como este los agrega a la base de datos.

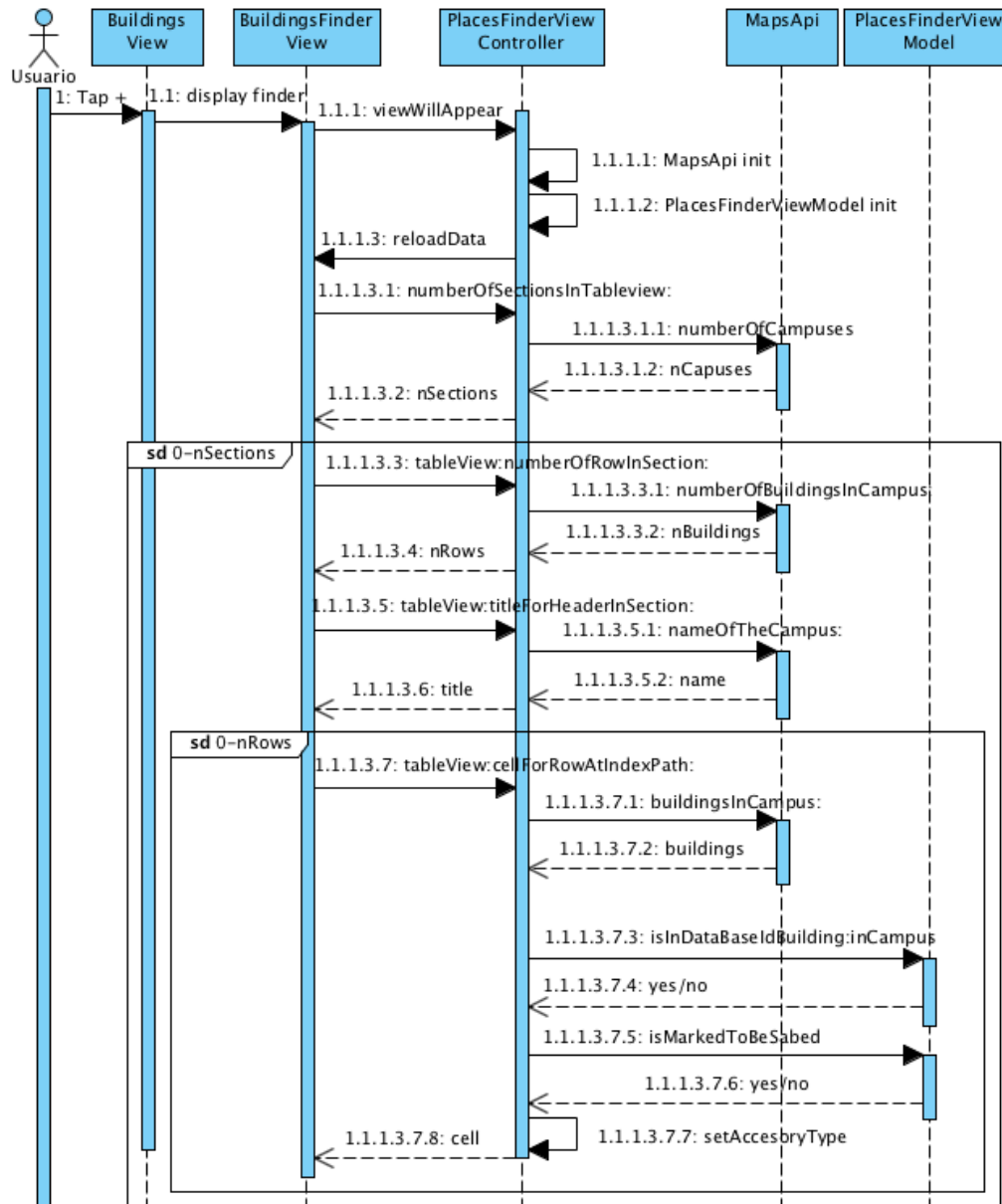
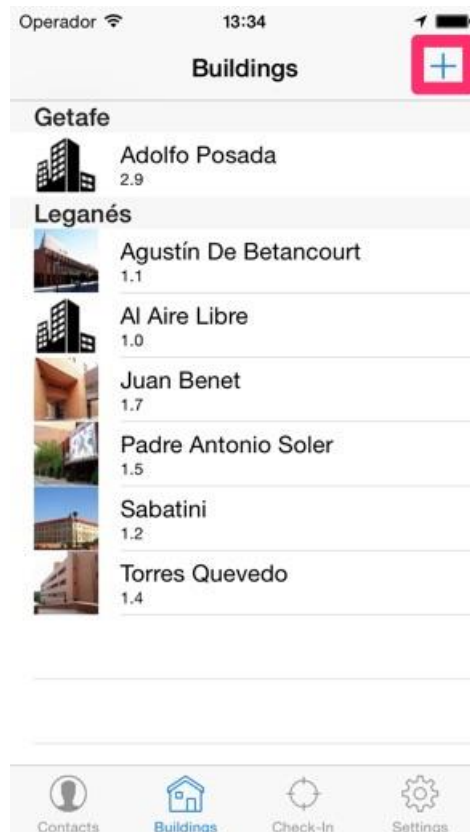


Diagrama de secuencia 10: Mostrar catálogo de edificios de la UC3M

El usuario accede a la búsqueda desde la pantalla de edificios. Cuando se carga la pantalla de búsqueda se inicializa la clase MapsApi y la PlacesFindesViewModel. La inicialización del MapsApi no es trivial, ya que requiere hacer las consultas necesarias para obtener los campus y los edificios. Esta es la razón por la que se llama al método reloadData del control cuanto termina el proceso para desencadenar la secuencia de pintado habitual para los UITableViews.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Inicio de secuencia 8: Tap +

A continuación el usuario debe seleccionar los edificios que desea añadir y pulsar el botón de guardado lo que desencadena las siguientes secuencias de eventos.

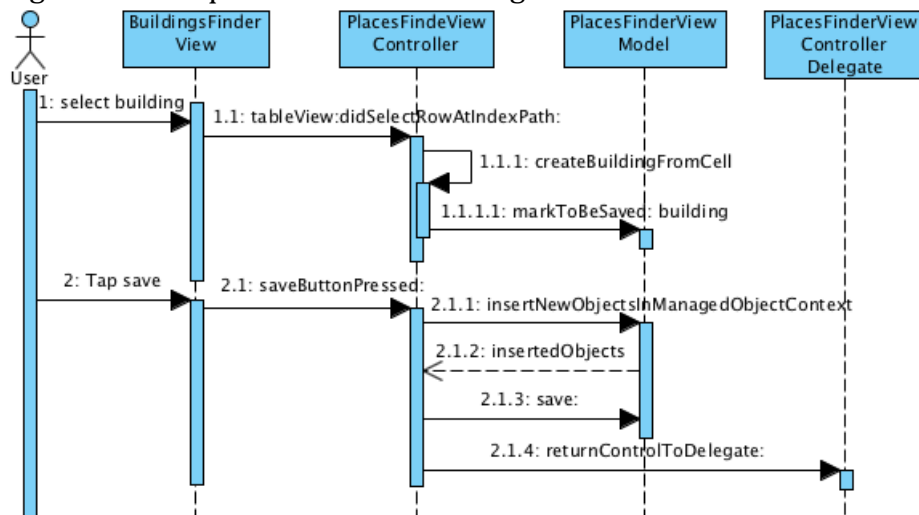
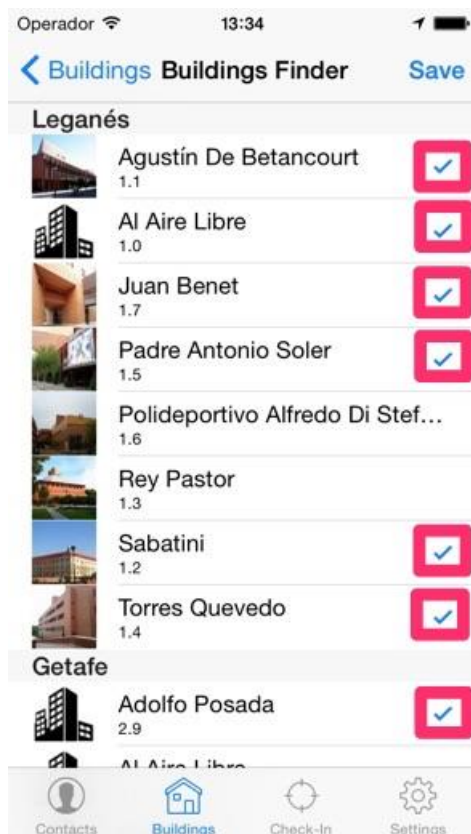


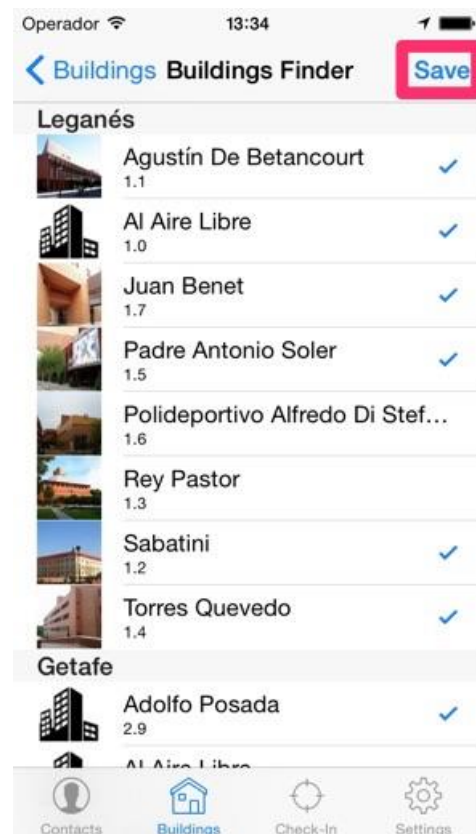
Diagrama de secuencia 11: Añadir edificios

El usuario inicia la secuencia seleccionando edificios de la lista. Cada selección del usuario provoca que se genere un diccionario con la información del edificio pulsado que se añade a una lista en el PlacesFinderViewModel mediante el método markToBeSaved. Esta lista de objetos se usará después para almacenar en la base de datos los edificios seleccionados por el usuario.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Inicio de secuencia 9: Select building



Inicio de secuencia 10: Tap Save

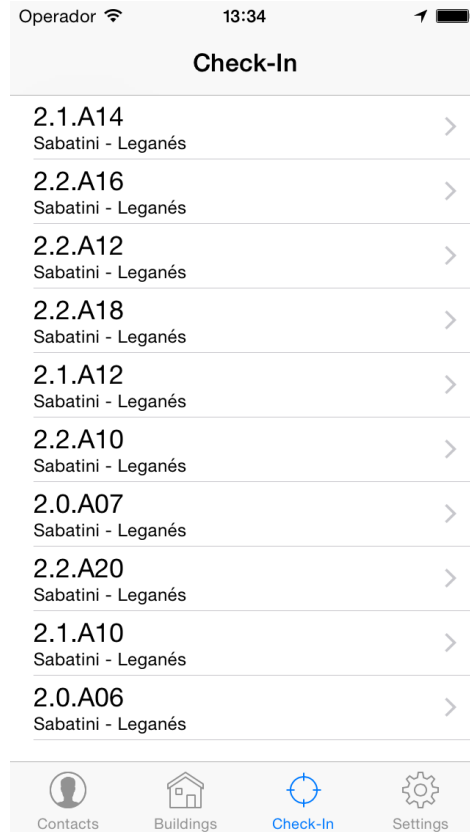
En el caso de que el usuario pulse sobre una celda ya marcada para ser guardada se hace uso del método `unmarkToBeSaved` para eliminar de la lista del `PlaceFinderViewModel` y que no sea guardado cuando el usuario pulse el botón de guardado.

Cuando el usuario decide guardar su selección pulsa el botón de guardado provocando la ejecución del método `saveButtonPressed`: del controlador. Este ordena al `PlaceFinderViewModel` que inserte los objetos de la lista en el contexto de la base de datos para después decirle que los persista con el método `save`. La lista de objetos insertados se usa para que la lea el delegado del control tras ejecutar el `returnControlToDelegate`, este deberá monitorizar las geo-fronteras de los nuevos edificios.

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

4.2.2.3 Checkins

Este módulo únicamente tiene la pantalla que lista los lugares cercanos al usuario en los que puede hacer checkin de forma manual, es por tanto donde se encuentra la implementación de la historia de usuario de Checkin manual.



La implementación de esta pantalla y su historia de usuario se distribuye en las clases que se muestran en el siguiente gráfico.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

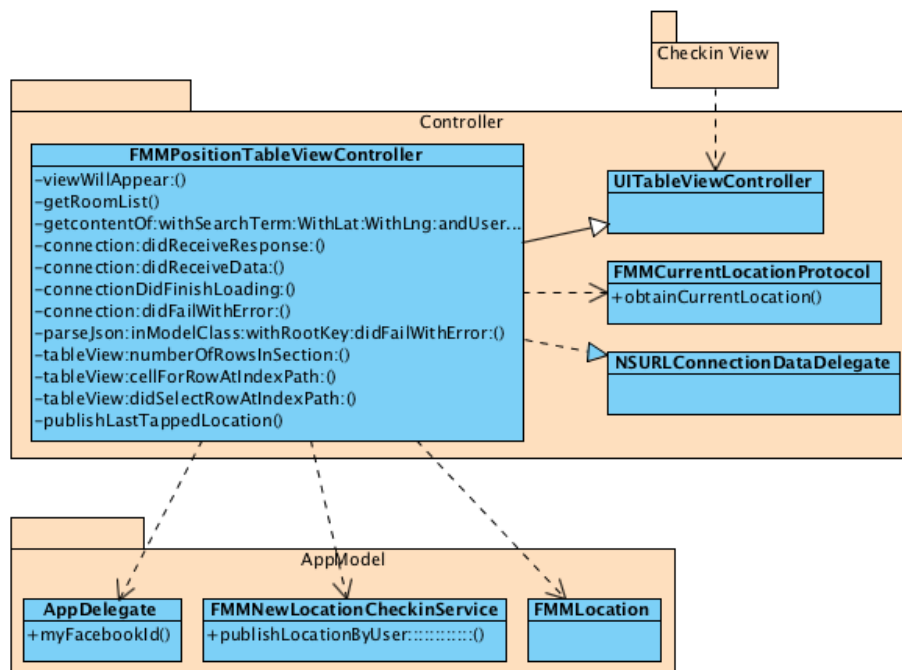


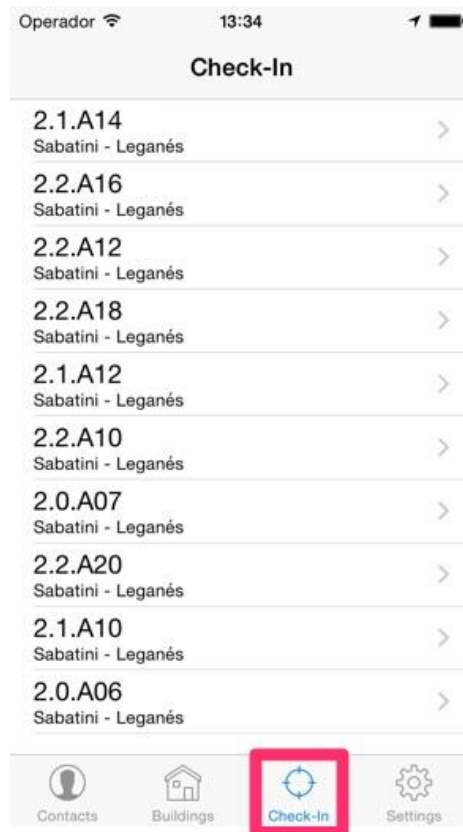
Diagrama de clases 8: Checkin manual

A continuación se muestra la secuencia de de eventos en la interacción de clases para ejecutar la historia de usuario.

Para que el usuario pueda elegir la posición cercana en la que se encuentra primero se genera el listado.

La secuencia se inicia cuando el usuario abre la pestaña de checkin. En este momento se ejecuta el método `getRomList` para inicializar la secuencia que acabará con la obtención de la lista de lugares cercanos al usuario. Para ello primero se obtiene la posición geográfica del usuario a partir de la lectura del GPS del teléfono haciendo uso del método `obtainCurrentLocation` de una clase que cumple con el protocolo `FMMCurrentLocationProtocol`.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Inicio de secuencia 11: Select checkin tab

Estas coordenadas se usan para consultar el repositorio de lugares de la UC3M. Esta se realiza usando el protocolo http por lo que usa la clase del framework de iOS `NSURLConnection` inicializando una conexión con un request al método "find" de la API de mapas usando como únicos parámetros no nulos las coordenadas actuales del usuario.

Tras la ejecución del request, el `FMMPositionTableViewController` como delegado de la conexión comienza a recibir los mensajes definidos por el protocolo para la gestión de la recepción de la respuesta al request.

Cuando la conexión termina y el sistema operativo ha pasado todos los mensajes a la aplicación se llama al método `connectionDidFinishLoading`; y libera al delegado. Es en este momento cuando se analiza el stream recibido como un string json para crear los objetos `FMMLocation` que serán representados en la tabla.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

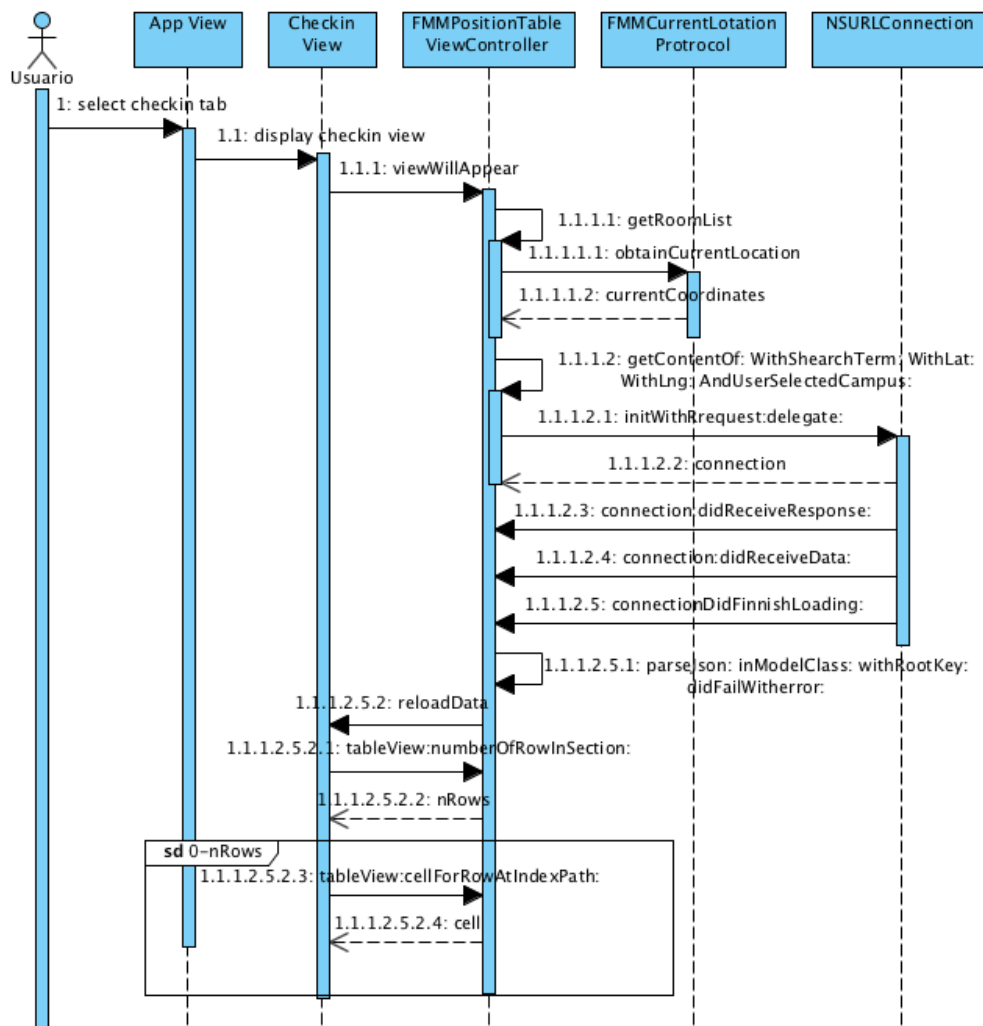


Diagrama de secuencia 12: Mostrado de lugares cercanos al usuario

Cuando el usuario selecciona una localización cercana de la tabla se presenta una alerta que pide confirmación. Entonces, si el usuario acepta, se publica el checkin del lugar que ha seleccionado.

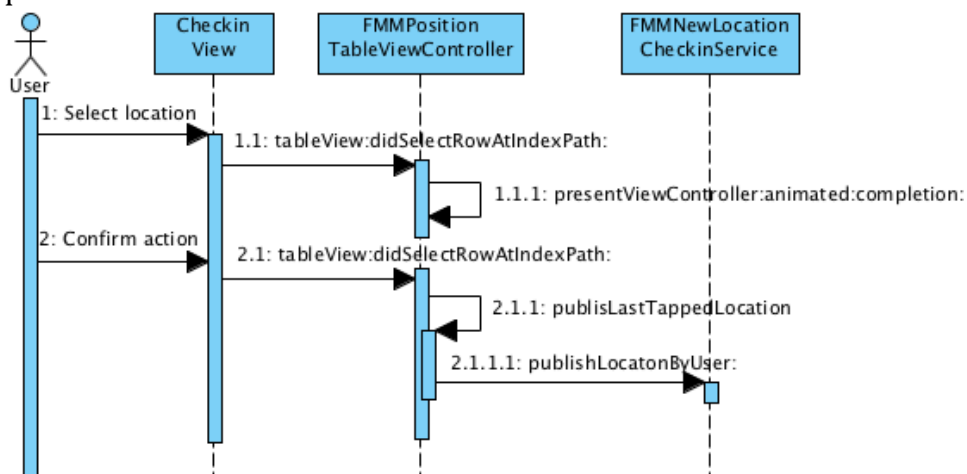
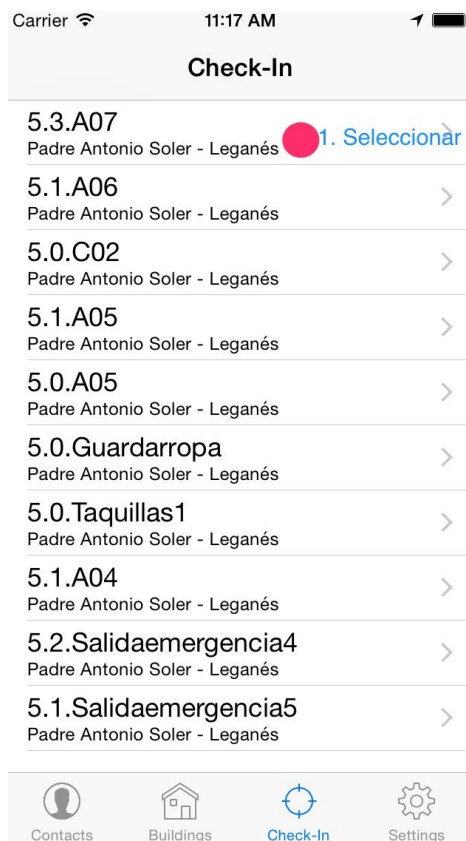
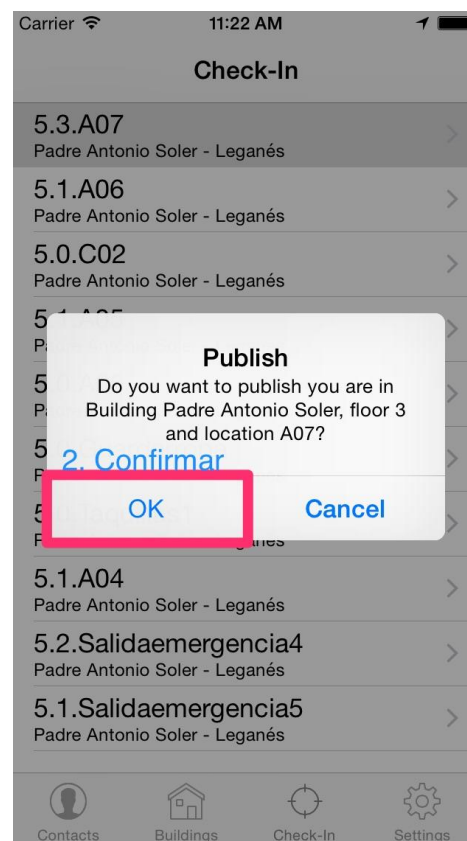


Diagrama de secuencia 13: Checkin manual

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Inicio de secuencia 12: Select location

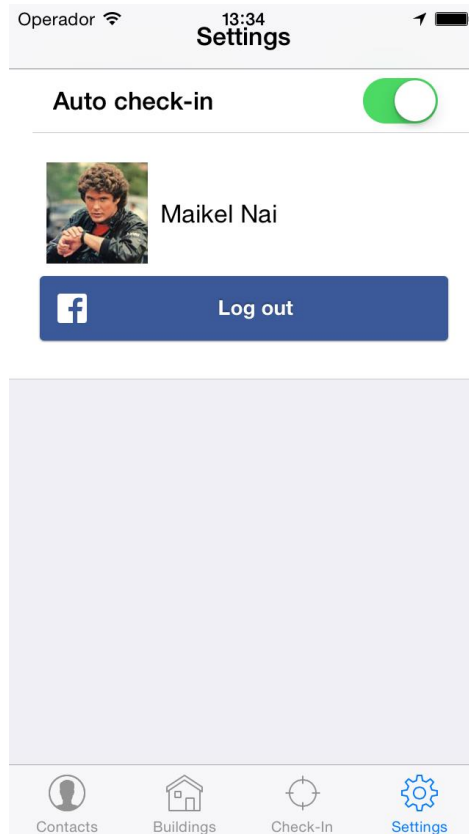


Inicio de secuencia 13: Confirm action

4.2.2.4 Settings

Este módulo únicamente tiene una vista para que el usuario configure active o desactive el autocheckin y se registre con su cuenta de Facebook.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



La implementación de esta pantalla y su historia de usuario se distribuye en las clases que se muestran en el siguiente diagrama.

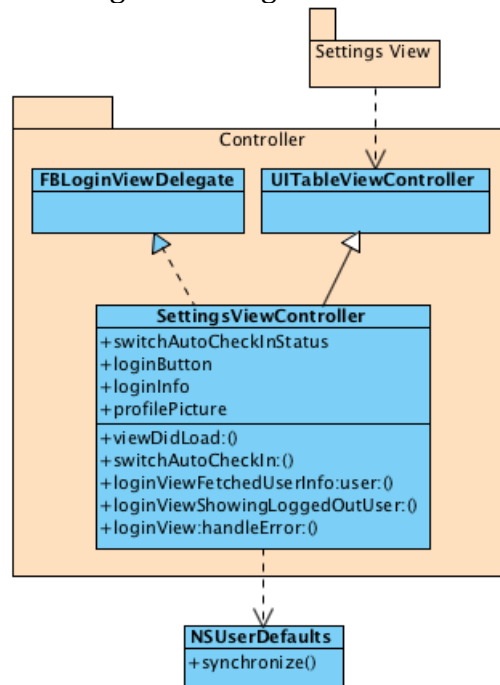


Diagrama de clases 9: Settings

FBLoginViewDelegate es un protocolo que se debe cumplir para controlar las respuestas de Facebook cuando se usa el botón FBLoginView que proporciona su sdk para hacer login o logout.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma iOS)

Del SDK de Facebook también se usa el control de usuario `FBProfilePictureView` para mostrar la imagen del perfil del usuario que se registra.

Para almacenar el identificador de usuario y el estado del auto checkin se usa la clase `NSUserDefaults` del framework de iOS.

El log in se implementa como se muestra en la siguiente secuencia.

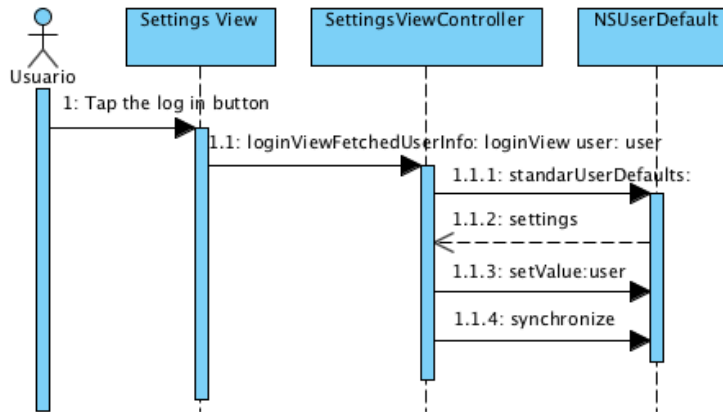


Diagrama de secuencia 14: Log in

El usuario inicia la secuencia en la pestaña de settings pulsando en el botón de login. Cuando el SDK de Facebook termina de hacer el log in sin errores cambia el botón para que sea el de log out y llama al método delegado `loginViewFetchedUserInfo:user:` en el que envía la información del usuario registrado en el parámetro `user`. Esta información es la que se usa para almacenar el identificador del usuario en las settings haciendo uso de la clase `NSUserDefaults`.



Inicio de secuencia 14: Tap the log in buton

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

A continuación se muestra el log out, Es muy similar al log in la diferencia está en que el método delegado usado es otro y el valor se borra de las settings.

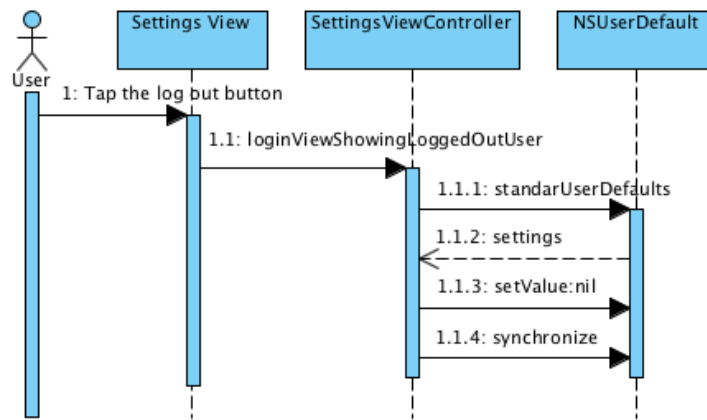


Diagrama de secuencia 15: Log out

El usuario pulsa el botón de log out y cuando el SDK de Facebook termina de ejecutar el log out llama al método delegado loginViewShowingLoggedOutUser:.



Inicio de secuencia 15: Tap the log out buton

En la implementación de este método se actualizan las NSDefault borrando el identificador de usuario.

A continuación se muestra la secuencia para desactivar el Auto checkin.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

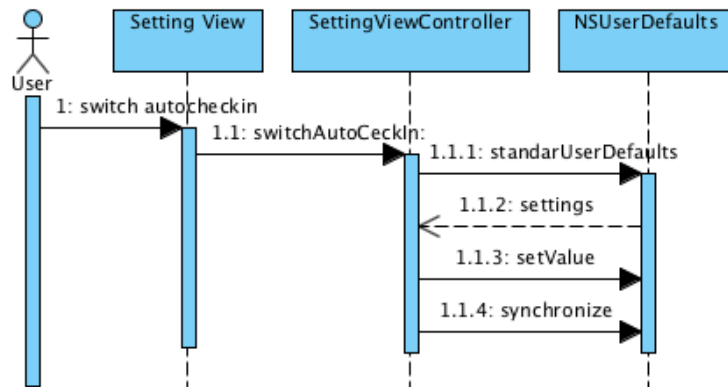


Diagrama de secuencia 16: Activar/Desactivar autocheckin

El usuario usa el interruptor de autocheckin para activar o desactivar la publicación automática de la llegada a los edificios del usuario.

En el método switchAutoCheckin se lee el valor del interruptor para guardarlo en las NSUserDefaults.



Inicio de secuencia 16: Switch autocheckin

4.2.2.5 App model

Es un modulo que forma parte del código del modelo de la aplicación. Este al ser común para el resto de módulos se ha colocado en un paquete aparte.

Este módulo, además, es el responsable de implementar la historia de usuario de autocheckin que no tiene interfaz de usuario al tratarse de una actividad autónoma.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

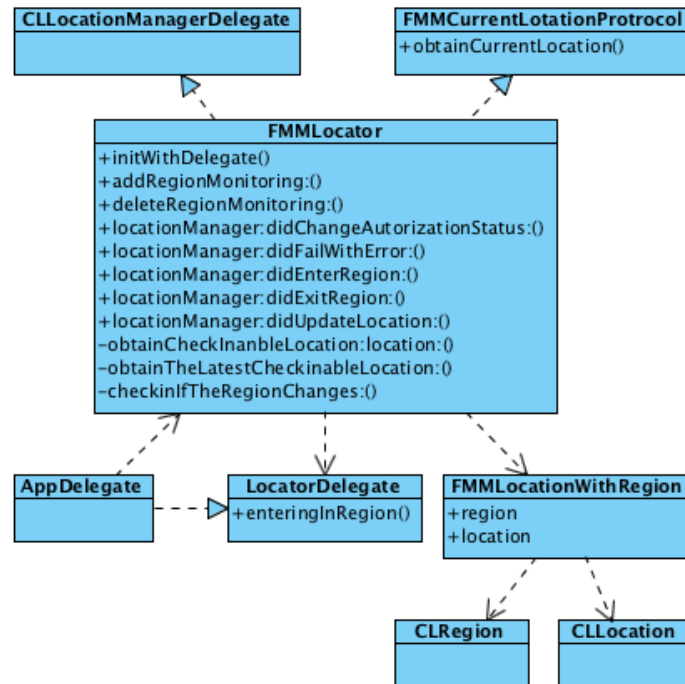


Diagrama de clases 10: Auto checkin

La clase **FMMLocator** implementa el protocolo **CLLocationManagerDelegate** y gestiona la autorización del usuario para que de su permiso para leer la posición del GPS.

Además implementa el protocolo **FMMCurrentLocationProtocol** para que cualquier otro módulo de la aplicación pueda conocer las coordenadas geográficas del teléfono si el usuario ha autorizado el uso del GPS.

A través del protocolo **LocatorDelegate** el **FMMLocator** avisa a su delegado que el teléfono ha entrado una geo-frontera.

Las geo-fronteras las añade al **FMMLocator** el **AppDelegate** cuando el usuario añade edificios.

FMMLocationWithRegion es una clase sin métodos que se usa para vincular una región delimitada por una geo-frontera con unas coordenadas leídas. Esto es, si se lee una posición (**CLLocation**) del GPS que se encuentra dentro de una de las regiones añadidas se crea este objeto que será de utilidad para determinar cuál fue la última región en la que estuvo el usuario tras un muestreo retardado.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

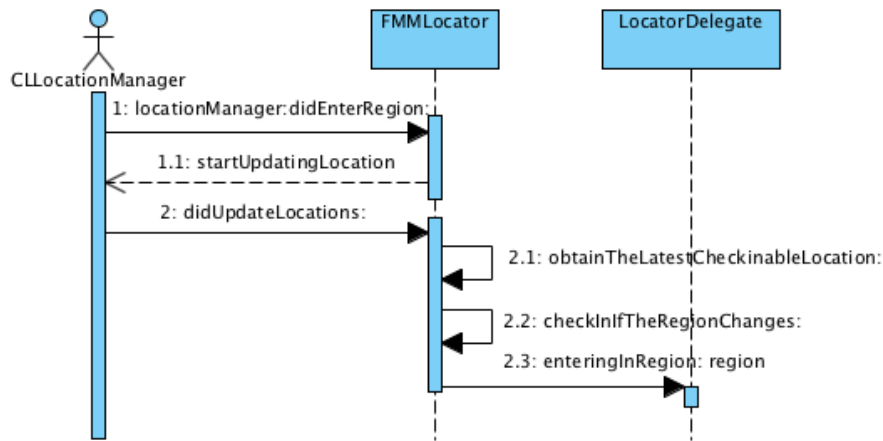


Diagrama de secuencia 17: Auto checkin

El locationManager inicia la secuencia al detectar que se entra en una región. Esta detección en realidad es una aproximación del sistema para ahorrar batería por lo que antes de emitir la notificación al delegado se comprueba si las coordenadas que se obtienen están en el interior de la región. Si es así se notifica al delegado, pero si no se inicia la monitorización de manera casi continuada hasta que se detecte que se sale de una región.

Para preservar la batería se configura el locationManager para que la precisión sea de diez metros, para que el didUpdateLocations lo ejecute cada 10 metros de desplazamiento del usuario y se le indica que la actividad es de tipo Fitnes. Con esto tenemos que si el usuario está quieto o se mueve dentro de un despacho, el GPS se encontraría en reposo. Además se activan las notificaciones retardadas ya que se considera que no se considera crítico publicar un checkin en tiempo real a favor del ahorro de batería.

Como las notificaciones retardadas están activadas al llamar a didUpdateLocations el locationManager pasa un array de localizaciones con sus marcas de tiempo.

ObtainTheLatestCheckinableLocation recorre este array y devuelve la última región (edificio) en la que estuvo el usuario y si esta se corresponde con una distinta a la del último checkin se emitirá una nueva notificación al LocatorDelegate avisando de que se ha entrado en una nueva región

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma
IOS)

Capítulo 5: Pruebas de la aplicación

- Contenido

5	PRUEBAS DE SISTEMA.....	95
5.1	US-01: VER CONTACTOS.....	95
5.2	US-02: AÑADIR AMIGO	95
5.3	US-03: ELIMINAR CONTACTO	97
5.4	US-04: VER EDIFICIOS DE AUTOCHECKIN	98
5.5	US-05 AÑADIR EDIFICIOS	98
5.6	US-06: ELIMINAR EDIFICIO	99
5.7	US-07: CHECKIN MANUAL	99
5.8	US-08: AUTOCHECKIN	100
5.9	US-09 DESACTIVAR AUTOCHECKIN	101
5.10	US-10: ACTIVAR AUTOCHECKIN	101
5.11	US-11: HACER LOGIN	101
5.12	US-12: HACER LOGOUT.....	102
5.13	US-13 VER CHECKINS.....	103
5.14	US-14: VER CHECKINS EN UN GIS.....	104
5.15	MATRIZ DE TRAZABILIDAD.....	104

Estas pruebas tratan de asegurar la correctitud del sistema. Se definen en una tabla con los siguientes campos:

- ID: Es el identificador de la prueba
- Descripción: Es el objetivo que se persigue probar
- Paso: La secuencia de la acción que se debe hacer con la aplicación para realizar la prueba.
- Acción: La acción que se debe realizar en la aplicación en el paso correspondiente
- Resultado esperado: Resultado que se espera de la aplicación para la acción realizada
- Errores: Causas que pueden provocar que no se pase la prueba.

ID		
Descripción		
Paso	Acción	Resultado esperado
Errores posibles		

Tabla 2: Plantilla de prueba de sistema

5.1 US-01: Ver contactos.

ID	PS-01	
Descripción		
La aplicación muestra los contactos agregados		
Paso	Acción	Resultado esperado
1	Con la aplicación iniciada abrir la pestaña de contactos	Se muestran los contactos del usuario
Errores posibles		

ID	PS-02	
Descripción		
Comprobar la navegabilidad de la aplicación permitiendo al usuario entrar en la pantalla que le permita añadir amigos y volver sin necesidad de haber añadido ninguno		
Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña “Contacts”.	La pestaña mostrada es la de los contactos y se muestran los contactos previamente añadidos.
2	En la barra de navegación pulsar el botón “+”	Se muestra una nueva pantalla en la que se listan todos los amigos de Facebook del

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

		usuario que no hay sido previamente añadidos al sistema.
3	En la barra de navegación pulsar el botón "< Contacts"	Se muestra el contenido de la pestaña de contactos listando los contactos previamente añadidos al sistema

Errores posibles

- No hay conexión a internet
- El usuario no ha iniciado sesión en Facebook

ID PS-03

Descripción

Comprobar que los amigos pueden añadirse al sistema

Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña de contactos.	La pestaña mostrada es la de los contactos y se muestran los contactos previamente añadidos.
2	En la barra de navegación pulsar el botón "+"	Se muestra una nueva pantalla en la que se listan todos los amigos de Facebook del usuario que no han sido previamente añadidos al sistema.
3	Seleccionar uno de los amigos del listado	La fila seleccionada queda marcada para ser añadida
4	Pulsar el botón "Save"	La pantalla retorna al listado de contactos mostrando el nuevo contacto añadido
5	Pulsar de nuevo sobre el botón "+" en la barra de navegación	Se muestra la pantalla en la que se listan todos los amigos de Facebook del usuario que no han sido añadidos previamente al sistema. El usuario añadido recientemente no se lista.
6	En la barra de navegación pulsar el botón "< Contacts"	Se muestra el contenido de la pestaña de contactos listando los contactos previamente añadidos al sistema

Errores posibles

- No hay conexión a internet
- El usuario no ha iniciado sesión en Facebook
- El servicio web del sistema no responde

ID PS-04

Descripción

Comprobar que los amigos pueden añadirse al sistema usando selección múltiple.

Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña de contactos.	La pestaña mostrada es la de los contactos y se muestran los contactos previamente añadidos.
2	En la barra de navegación pulsar el botón "+"	Se muestra una nueva pantalla en la que se listan todos los amigos de Facebook del usuario que no han sido previamente añadidos al sistema.
3	Seleccionar todos los amigos del listado	Las filas seleccionadas quedan marcadas para ser añadidas

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

4	Pulsar el botón "Save"	La pantalla retorna al listado de contactos mostrando los nuevos contactos añadidos
5	Pulsar de nuevo sobre el botón "+" en la barra de navegación	Se muestra la pantalla den la que se listan todos los amigos de Facebook del usuario que no han sido añadidos previamente al sistema. El listado se muestra vacío.
6	En la barra de navegación pulsar el botón "< Contacts"	Se muestra el contenido de la pestaña de contactos listando los contactos previamente añadidos al sistema

Errores posibles

- No hay conexión a internet
- El usuario no ha iniciado sesión en Facebook
- El servicio web del sistema no responde

5.3 US-03: Eliminar contacto

ID	PS-05	
Descripción		
Comprobar que los amigos pueden eliminarse del sistema.		
Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña de contactos.	La pestaña mostrada es la de los contactos y se muestran los contactos previamente añadidos.
2	Sobre el contacto que se desea eliminar, deslizar la celda hacia la izquierda	En el lado derecho de la celda aparece un cuadro que pide confirmación para eliminar.
3	Pulsar “Delete”	El usuario desaparece del listado
5	Pulsar de nuevo sobre el botón “+” en la barra de navegación	Se muestra la pantalla den la que se listan todos los amigos de Facebook del usuario que no han sido añadidos previamente al sistema. En el listado puede verse el usuario recientemente eliminado.
6	En la barra de navegación pulsar el botón “< Contacts”	Se muestra el contenido de la pestaña de contactos listando los contactos previamente añadidos al sistema
Errores posibles		
<ul style="list-style-type: none">• No hay conexión a internet• El servicio web del sistema no responde		

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

5.4 US-04: Ver edificios de autocheckin

ID PS-06

Descripción

Comprobar que se muestra el listado de edificios añadidos al sistema para que este haga autocheckin de forma autónoma cuando se entra en alguno de ellos.

Paso **Acción**

Resultado esperado

1	Con la aplicación iniciada, acudir a la pestaña "Buildings".	La pestaña mostrada es la de los edificios y se muestran los edificios previamente añadidos.
---	--	--

Errores posibles

5.5 US-05 Añadir edificios

ID PS-07

Descripción

Comprobar la navegabilidad de la aplicación permitiendo que el usuario entre en la pantalla de añadir edificios y pueda salir sin tener que añadir ninguno.

Paso **Acción**

Resultado esperado

1	Con la aplicación iniciada, acudir a la pestaña "Buildings".	La pestaña mostrada es la de los edificios y se muestran los edificios previamente añadidos.
2	En la barra de navegación pulsar el botón "+"	Se muestra una nueva pantalla donde se listan los edificios de la UC3M agrupados por campus. Los edificios previamente añadidos aparecen marcados.
3	En la barra de navegación pulsar "< Buildings"	La pantalla mostrada lista los edificios previamente añadidos al sistema.

Errores posibles

- No hay conexión a internet
- El servicio de mapas de UC3M No responde

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

ID	PS-08	
Descripción		
Comprobar que los edificios ofrecidos por el servicio web de la universidad pueden añadirse al sistema.		
Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña “Buildings”.	La pestaña mostrada es la de los edificios y se muestran los edificios previamente añadidos.
2	En la barra de navegación pulsar el botón “+”	Se muestra una nueva pantalla donde se listan los edificios de la UC3M agrupados por campus. Los edificios previamente añadidos aparecen marcados.
3	Seleccionar una fila	La fila seleccionada aparece marcada para ser guardada
4	En la barra de navegación pulsar “Save”	La aplicación retorna a la pantalla “Buildings” y muestra un listado con los edificios que han sido añadidos al sistema
5	En la barra de navegación pulsar el botón “+”	Se muestra una nueva pantalla donde se listan los edificios de la UC3M agrupados por campus. Los edificios previamente añadidos aparecen marcados.
Errores posibles		
<ul style="list-style-type: none">• No hay conexión a internet• El servicio de mapas de UC3M No responde		

5.6 US-06: Eliminar edificio

ID	PS-09	
Descripción		
Comprobar que los edificios pueden eliminarse del sistema.		
Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña “Buildings”.	La pestaña mostrada es la de los edificios en los que el sistema hace checkin de manera autónoma. Se muestran los edificios añadidos previamente.
2	Sobre el edificio que se desea eliminar, deslizar la celda hacia la izquierda	En el lado derecho de la celda aparece un cuadro que pide confirmación para eliminar.
3	Pulsar “Delete”	El edificio desaparece del listado
5	Pulsar de nuevo sobre el botón “+” en la barra de navegación	Se muestra la pantalla den la que se listan todos los edificios de la UC3M. En el listado puede verse que el edificio recientemente eliminado no está marcado como añadido.
6	En la barra de navegación pulsar el botón “< Buildings”	Se muestra el contenido de la pestaña de contactos listando los edificios previamente añadidos al sistema
Errores posibles		

5.7 US-07: Checkin manual

ID	PS-10
-----------	--------------

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Descripción

Comprobar que el usuario puede hacer manualmente un checkin en las ubicaciones cercanas que son proporcionadas por el servicio web de mapas de la UC3M.

Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña "Checkin".	La pestaña mostrada lista un conjunto de lugares cercanos al usuario.
2	Seleccionar un lugar	Aparece una pantalla de confirmación.
3	Aceptar	El lugar es publicado. Se vuelve al estado de la pantalla anterior.
		Los contactos que tenga agregados al usuario y que este también tenga agregado al contacto verán en la ventana de contactos y en el GIS cual ha sido la última posición publicada del usuario.

Errores posibles

- No hay conexión a internet
- El servicio de mapas de UC3M no responde
- El servicio web del sistema no responde

ID PS-11

Descripción

Comprobar que el sistema permite cancelar la acción de hacer manualmente un checkin en las ubicaciones cercanas que son proporcionadas por el servicio web de mapas de la UC3M.

Paso	Acción	Resultado esperado
1	Con la aplicación iniciada, acudir a la pestaña "Checkin".	La pestaña mostrada lista un conjunto de lugares cercanos al usuario.
2	Seleccionar un lugar	Aparece una pantalla de confirmación.
3	Cancelar	El lugar es publicado. Se vuelve al estado de la pantalla anterior.
		Los contactos que tenga agregados al usuario y que este también tenga agregado al contacto no verán en la ventana de contactos ni en el GIS cual ha sido la última posición publicada del usuario.

Errores posibles

- No hay conexión a internet
- El servicio de mapas de UC3M no responde
- El servicio web del sistema no responde

5.8 US-08: Autocheckin

ID PS-12

Descripción

Comprobar que el sistema publica a los amigos del usuario que se encuentra en un edificio de los añadidos al cruzar la frontera geográfica que lo delimita.

Paso	Acción	Resultado esperado
1	Con la aplicación ejecutándose en segundo plano entrar en un edificio previamente añadido a la aplicación.	El sistema emite la publicación

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Los contactos que tenga agregados al usuario y que este también tenga agregado al contacto verán en la ventana de contactos y en el GIS cual ha sido la última posición publicada del usuario.

Errores posibles

- No hay conexión a internet
- El servicio web del sistema no responde

5.9 US-09 Desactivar autocheckin

ID PS-13

Descripción

Comprobar que tras desactivar el autocheckin el sistema deja de publicar si el usuario se encuentra en uno de los edificios añadidos.

Paso	Acción	Resultado esperado
1	Con la aplicación abierta y configurada para hacer autocheckin abrir la ventana de configuración seleccionando la pestaña "Settings".	La pantalla de configuración se muestra. El interruptor "Auto check-in" está encendido.
2	Apagar el interruptor "Auto check-in". A continuación entrar en uno de los edificios añadidos.	Los contactos del usuario que hayan sido añadidos no verán una actualización en el estado del usuario.

Errores posibles

5.10 US-10: Activar autocheckin

ID PS-14

Descripción

Comprobar que tras activar el autocheckin el sistema comienza a publicar si el usuario se encuentra en uno de los edificios añadidos.

Paso	Acción	Resultado esperado
1	Con la aplicación abierta y configurada para no hacer autocheckin abrir la ventana de configuración seleccionando la pestaña "Settings".	La pantalla de configuración se muestra. El interruptor "Auto check-in" está apagado.
2	Encender el interruptor "Auto check-in". A continuación entrar en uno de los edificios añadidos.	Los contactos del usuario que hayan sido añadidos verán una actualización en el estado del usuario.

Errores posibles

5.11 US-11: Hacer login

ID PS-15

Descripción

Comprobar que el usuario puede iniciar sesión con su cuenta de Facebook.

Paso	Acción	Resultado esperado
------	--------	--------------------

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

1	Con la aplicación abierta y con la sesión de usuario cerrada en FMM abrir la ventana de contactos seleccionando la pestaña "Contacts"	Se muestra un listado con los amigos añadidos previamente
2	En barra de navegación hacer click en el botón "+"	Se avisa al usuario que no está conectado al Facebook y se muestra el listado de amigos vacío
3	Cerrar el aviso y volver a la pantalla de contactos pulsando en la barra de navegación sobre "< Contacts"	La ventana de contactos lista los amigos añadidos previamente.
4	Abrir la ventana de configuración seleccionando la pestaña "Settings".	La pantalla de configuración se muestra. Se muestra una silueta y un botón para iniciar sesión con Facebook.
5	Pulsar el botón "Log in with Facebook".	El sistema pide las credenciales de Facebook y autorización para que FMM acceda a la información básica del usuario que se acredita. Si el usuario tiene ya una sesión iniciada en el terminal, FMM recupera las credenciales pidiendo únicamente autorización.
6	En caso necesario introducir las credenciales y autorizar a FMM. En caso de haber sido autorizada la aplicación con anterioridad tampoco será necesaria confirmar la autorización.	Se retorna a la ventana de "Settings". En lugar de la silueta se muestra la foto de perfil del usuario y el botón cambia a "Log out". A partir de este momento es posible hacer checkin y autocheckin
7	Abrir la ventana de contactos seleccionando la pestaña "Contacts"	La ventana de contactos muestra los usuarios añadidos previamente.
8	En la barra de navegación pulsar en el botón "+"	Se muestra una pantalla en la que se listan los amigos de Facebook del usuario que tienen FMM instalada.

Errores posibles

- No hay conexión a internet

5.12 US-12: Hacer logout

ID	PS-16	
Descripción		
Comprobar que el usuario puede cerrar la sesión con su cuenta de Facebook.		
Paso	Acción	Resultado esperado
1	Con la aplicación abierta y con la sesión de usuario iniciada en FMM abrir la ventana de contactos seleccionando la pestaña “Contacts”	Se muestra un listado con los amigos añadidos previamente
2	En barra de navegación hacer click en el botón “+”	Se muestra el listado de amigos que usan FMM y no han sido añadidos aún a los contactos de FMM.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

3	Volver a la pantalla de contactos pulsando en la barra de navegación sobre "< Contacts"	La ventana de contactos lista los amigos añadidos con anterioridad.
4	Abrir la ventana de configuración seleccionando la pestaña "Settings".	La pantalla de configuración se muestra. Se muestra la foto de perfil de Facebook del usuario y un botón para cerrar sesión con Facebook.
5	Pulsar el botón "Log out".	El sistema pide confirmación para cerrar la sesión.
6	Aceptar.	Se retorna a la ventana de "Settings". En lugar de la foto se muestra una silueta y el botón cambia a "Login with Facebook". A partir de este momento no es posible hacer checkin y autocheckin
7	Abrir la ventana de contactos seleccionando la pestaña "Contacts"	La ventana de contactos muestra los usuarios añadidos previamente.
8	En la barra de navegación pulsar en el botón "+"	Se muestra una pantalla en la que no se listan los amigos de Facebook del usuario que tienen FMM instalada y un mensaje alerta de que no está iniciada la sesión con Facebook.

Errores posibles

- No hay conexión a internet

5.13 US-13 Ver checkins

ID	PS-17	
Descripción		
Comprobar que el usuario puede ver los checkins de sus contactos. Se debe comprobar que se cumple la regla que dos usuarios se tienen que añadir mutuamente a FMM para que puedan compartir la ubicación entre ellos.		
Paso	Acción	Resultado esperado
1	Con la aplicación abierta acudir a la ventana de contactos seleccionando la pestaña “Contacts”	Se muestra un listado con los amigos añadidos previamente
		Junto al nombre de los usuarios que han hecho checkin manual o autocheckin aparece la información del lugar y la hora del checkin
Errores posibles		
	<ul style="list-style-type: none">• No hay conexión a internet• El servicio web del sistema no responde	

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

5.14 US-14: Ver checkins en un GIS

ID	PS-18	
Descripción		
Comprobar que el usuario puede ver los checkins de sus contactos en un GIS. Se debe comprobar que se cumple la regla que dos usuarios se tienen que añadir mutuamente a FMM para que puedan compartir la ubicación entre ellos.		
Paso	Acción	Resultado esperado
1	Con la aplicación abierta acudir a la ventana de contactos seleccionando la pestaña “Contacts”	Se muestra un listado con los amigos añadidos previamente
		Junto al nombre de los usuarios que han hecho checkin manual o autocheckin aparece la información del lugar y la hora del checkin
2	En la barra de navegación pulsar sobre “Map”	Se muestra una pantalla con un GIS en el que se señala donde se encuentran los usuarios con checkins.
Errores posibles		
<ul style="list-style-type: none">• No hay conexión a internet• El servicio web del sistema no responde		

5.15 Matriz de trazabilidad

En este apartado se relacionan las historias de usuario con las pruebas del sistema que los cubren. Se distribuyen horizontalmente las historias de usuario y verticalmente las pruebas aplicadas.

	Us - 01	Us - 02	Us - 03	Us - 04	Us - 05	Us - 06	Us - 07	Us - 08	Us - 09	Us - 10	Us - 11	Us - 12	Us - 13	Us - 14
PS - 01	X													
PS - 02		X												
PS - 03		X												
PS - 04		X												
PS - 05			X											

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

PS - 06				X										
PS - 07					X									
PS - 08					X									
PS - 09						X								
PS - 10							X							
PS - 11								X						
PS - 12									X					
PS - 13										X				
PS - 14											X			
PS - 15												X		
PS - 16													X	
PS - 17														X
PS - 18														X

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Capítulo 6: Conclusiones y líneas futuras

Contenido

6	CONCLUSIONES	108
6.1	LÍNEAS FUTURAS	109
6.1.1	<i>Repositorios de información geográfica</i>	<i>109</i>
6.1.2	<i>Redes sociales.....</i>	<i>109</i>
6.1.3	<i>Privacidad</i>	<i>109</i>
6.1.4	<i>Idiomas</i>	<i>109</i>
6.1.5	<i>Integración.....</i>	<i>109</i>
6.1.6	<i>QR checkin</i>	<i>109</i>
6.1.7	<i>Localizaciones en interiores.....</i>	<i>109</i>
6.1.8	<i>Check-outs</i>	<i>109</i>
6.1.9	<i>Notificaciones.....</i>	<i>109</i>

6 Conclusiones

Actualmente existe un creciente interés por conocer el entorno que nos rodea y su estado. Cada vez las personas estamos más interesados en que hacen y qué interesa a nuestros amigos y compañeros para obtener información de ello. Nos interesan los lugares que visitan y sus opiniones y de los lugares nos interesan sus horarios, sus tarifas y las valoraciones que hacen de él sus usuarios o clientes.

El uso de los sistemas de información facilita que los usuarios gestionen toda esta información favoreciendo el crecimiento de la misma por lo que este proyecto se centra en ello tratando de ampliar la información y mejorar las capacidades que ofrece la aplicación de mapas de la UC3M.

La aplicación de mapas de la UC3M mantiene una relación entre un trabajador y su despacho pero no aporta la información de si el trabajador se encuentra en su despacho o está ausente. Este proyecto surge como una posible línea de aplicación para cubrir esta falta de información añadiendo además posicionamiento para todos los miembros de la comunidad universitaria.

Para el desarrollo se han usado las capacidades del terminal para conectarse a internet y detectar fronteras geográficas haciendo uso de los servicios del sistema operativo para acceder a coordenadas del GPS. Con la conectividad a internet podemos acceder a los amigos de Facebook, ver donde se encuentran los contactos, publicar donde se encuentra el usuario y mostrar la vista que proporciona GoogleMaps. Con los servicios de posicionamiento del sistema operativo podemos crear áreas geográficas que serán observadas para notificar a la aplicación cuando se entra en una de estas áreas. Y con el acceso a las coordenadas de GPS se puede añadir precisión a la localización, obtener lugares cercanos al usuario para hacer checkin manual y mostrar la posición del usuario en el mapa.

El desarrollo del proyecto ha resultado para mi muy satisfactorio al haberme permitido experimentar estas tecnologías que no conocía con anterioridad. He conocido detalles de cómo puede ser usado Facebook para el desarrollo de aplicaciones, también detalles sobre el SDK de Google maps y sobretodo como desarrollar una aplicación para iOS. Además el desarrollo de servicio web desplegado en Google App Engine me ha servido para probar la plataforma y aprender algo de Python que nunca lo había usado en otros proyectos.

Todo esto se traduce en dificultades para dar pasos durante el desarrollo pero me aporta un interesante background de conocimiento que me han permitido adquirir nuevas habilidades que considero importantes para continuar mi carrera profesional.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

6.1 Líneas futuras

6.1.1 Repositorios de información geográfica

La integración con Google places o cualquier otro repositorio podría ser también interesante. No sólo para añadir información de los distintos lugares de la uc3m, sino también para publicar eventos en los mismos o tener los emplazamientos alrededor del campus en los que también se desarrolla la vida universitaria.

Parte de las tareas universitarias se realizan también en otros campus de otras universidades, o en centros que no gestiona la UC3M por lo que permitir el acceso a otros repositorios distintos del de UC3M Mapas puede resultar de utilidad.

6.1.2 Redes sociales

Permitir que puedan buscarse usuarios en otras redes sociales eliminaría restricciones y facilitaría que más usuarios participen del uso de esta aplicación.

6.1.3 Privacidad

La aplicación tiene un login y un logout the Facebook. Si se quiere que la aplicación valga para más de un usuario deben asociarse a nivel de dominio de FMM los contactos con el usuario que tiene la sesión iniciada y los edificios añadidos. Hasta entonces los contactos de un usuario aparecen en el listado de otro y el sistema hará checkin automático en edificios que ha añadido otro usuario. La aplicación está desarrollada para un dispositivo considerado personal por lo que este caso de uso no se ha visto prioritario.

También se deben autorizar las llamadas a la API del servicio web para garantizar que el usuario que hace las peticiones está autenticado y tiene permisos para hacerlas.

6.1.4 Idiomas

Añadir la posibilidad de que el usuario elija en que idioma prefiere tener la aplicación puede ser una comodidad para los miembros de la comunidad que se encuentren disfrutando de un programa de intercambio en nuestra universidad.

6.1.5 Integración

Existe una aplicación similar de la UC3M que persigue los mismos objetivos que esta. Como línea futura se ve de utilidad que las dos aplicaciones compartan, desde el punto de vista del usuario, el almacén de datos para que las ubicaciones publicadas con la aplicación de Android puedan verse en la aplicación de iOS y viceversa.

También pudiera ser interesante unir las funcionalidades de la aplicación UC3M Mapas y esta en una sola aplicación.

6.1.6 QR checkin

Integrar la lectura de códigos QR para hacer los checkins manuales.

6.1.7 Localizaciones en interiores

Publicar de manera autónoma posiciones interiores. Es necesario dotar de infraestructura el espacio como el uso de iBeacons.

6.1.8 Check-outs

Que los usuarios publiquen cuando han abandonado un lugar es interesante porque añade la información de que ya no se encuentra en el sitio donde hizo el checkin.

6.1.9 Notificaciones

Puede ser interesante que el usuario pudiera configurar la aplicación para que le avise cuando uno de sus contactos llega o se va de una localización.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma
IOS)

ANEXO I – Manual de usuario

Contenido

ANEXO I: MANUAL DE USUARIO	112
REQUISITOS DE FUNCIONAMIENTO	112
PRIMEROS PASOS.....	113
<i>Permisos de localización</i>	113
<i>Login</i>	114
AJUSTES.....	115
CONTACTOS.....	116
<i>Añadir contactos</i>	116
<i>Borrar contactos</i>	117
<i>Mapa</i>	117
EDIFICIOS	119
<i>Añadir edificios</i>	119
<i>Eliminar edificios</i>	120
CHECKIN	120

Anexo I: Manual de Usuario

Requisitos de funcionamiento

Esta aplicación se ha desarrollado para iPhone 5 con sistema operativo iOS 8.1, siendo compatible con versiones superiores.

Para poder ser instalada es necesario de disponer de al menos 16MB libres y es recomendado también que el usuario tenga instalado Facebook en el terminal para tener mejor experiencia de usuario en las tareas de registro. Es por tanto esencial que el usuario disponga de cuenta en Facebook para usar esta aplicación.

La gran mayoría de las tareas que se pueden hacer con esta aplicación requieren el uso de una conexión activa a internet, ya sea mediante la conectividad wifi o la red de telefonía móvil. Además la monitorización de la posición del usuario es necesaria para hacer checkins automáticos y ofrecer los lugares correctos al usuario para que haga checkins manuales.

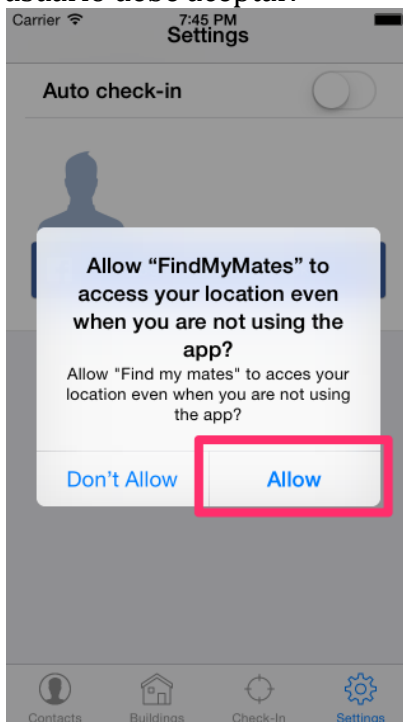
Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Primeros pasos

Para que la aplicación tenga el funcionamiento deseado es necesario que el usuario de permiso para que la aplicación use el GPS aunque la aplicación no esté activa y que haga login con su cuenta de Facebook aceptado la política de privacidad. Normalmente esto sucede la primera vez que el usuario abre la aplicación al mostrarse la pestaña de ajustes.

Permisos de localización

La primera vez que la aplicación necesite acceder a la información de localización del usuario se mostrará un mensaje como el de la imagen. Para que funcione el autocheckin y la aplicación sugiera lugares cercanos en la pestaña check-in el usuario debe aceptar.

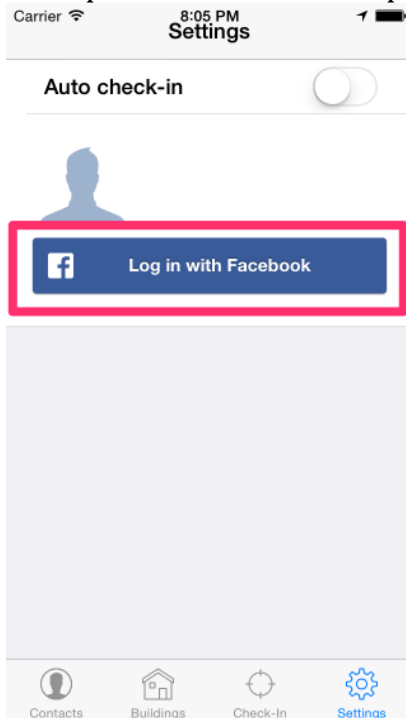


Pantalla 8: Permiso de acceso al GPS

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Login

En la pestaña de Settings el usuario puede iniciar sesión con su cuenta de facebook. Esta operación es necesaria para poder publicar los checkins.



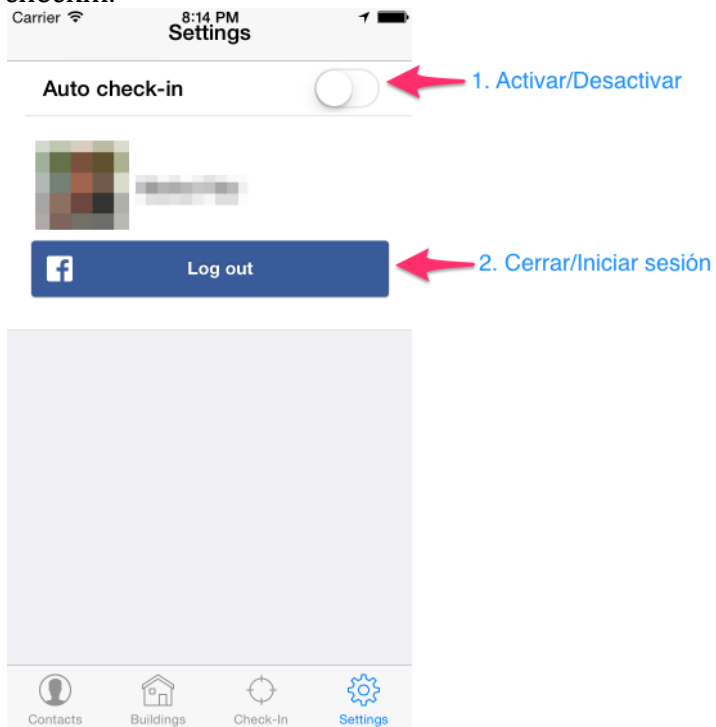
Pantalla 9: Registrarse en la App

Al pulsar el botón señalado se solicitan las credenciales de facebook y se pide que se acepte la política de privacidad para los datos que se da permiso leer a la aplicación. Tras aceptar, en esta pantalla debe aparecer la imagen de perfil y el nombre del usuario acreditado.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Ajustes

En esta pestaña el usuario puede hacer “log out”, log in activar y desactivar el auto checkin.

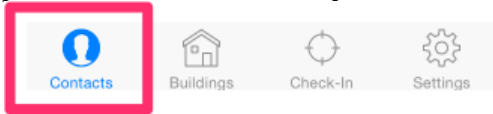


Pantalla 10: Ajustes

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Contactos

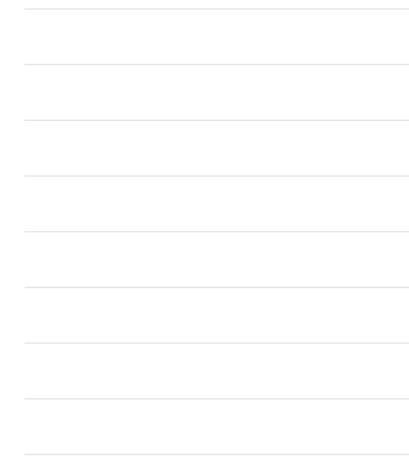
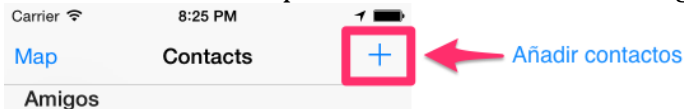
En la barra de pestañas que se encuentra visible en todo momento hay un icono que permite el acceso a esta pantalla.



11: Barra de pestañas

Añadir contactos

Desde la pantalla principal de contactos se pueden añadir amigos que tenga el usuario en facebook pulsando en la barra de navegación el botón +



Pantalla 12: Contactos

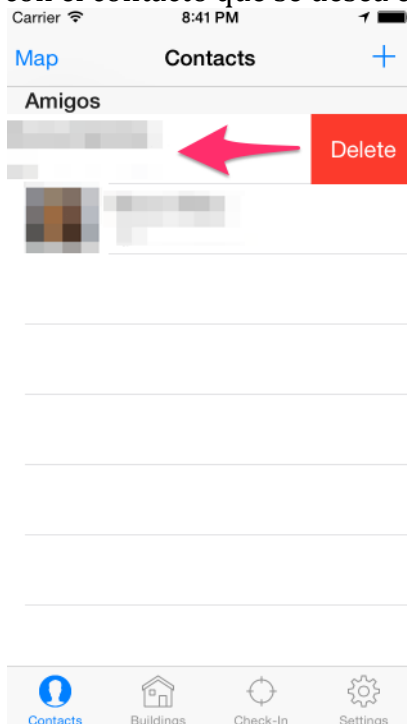
Tras la pulsación se transita a una pantalla donde se listan los amigos de Facebook que tienen la aplicación instalada.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Borrar contactos

Para borrar un contacto basta con desplazar a la izquierda la celda correspondiente con el contacto que se desea eliminar y pulsar el botón "Delete" que se muestra



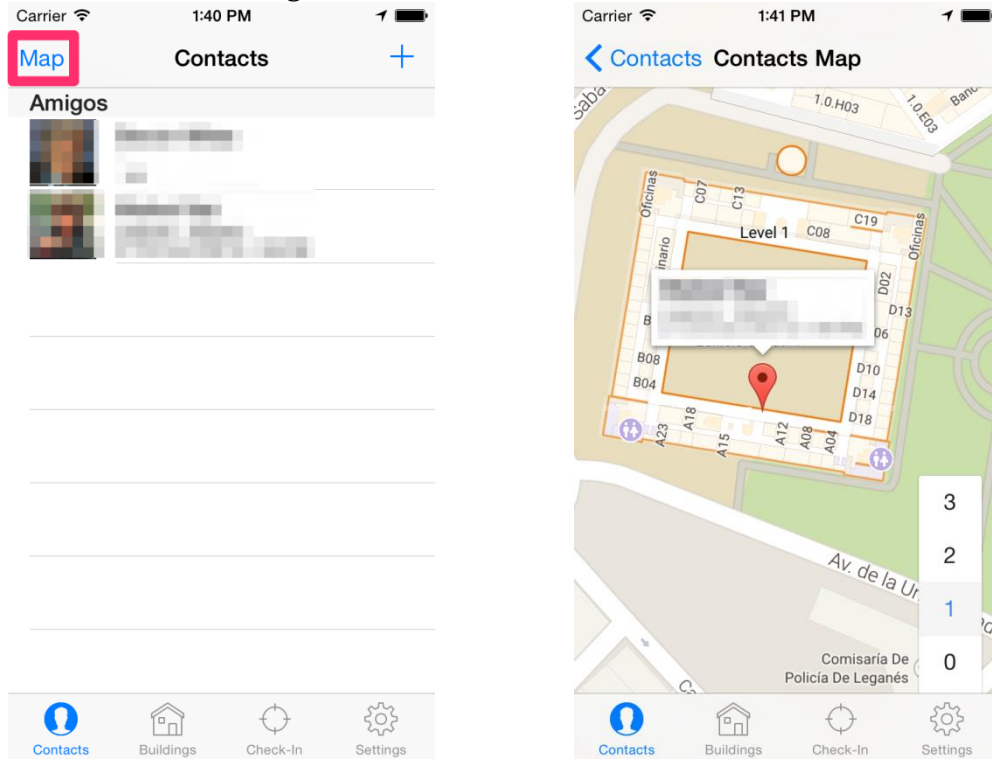
Pantalla 14: Eliminar contacto

Mapa

En la pantalla de mapa se puede ver la posición de los contactos sobre el mapa de Google Maps.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Para ello en la pantalla donde se listan los contactos es necesario pulsar el botón “Map” en la barra de navegación.

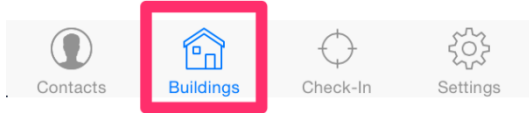


Pantalla 15: Mapa de checkins

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Edificios

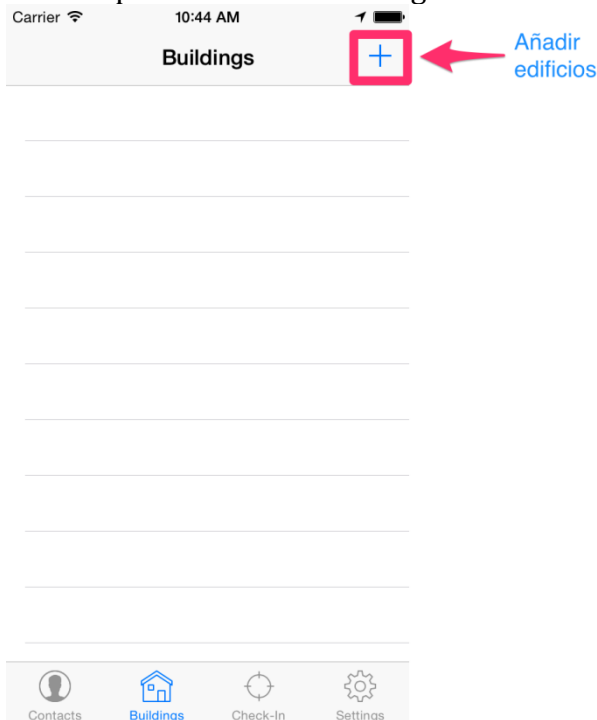
En esta pestaña se pueden añadir y borrar edificios en los que el dispositivo compartirá la presencia del usuario de forma autónoma. Podemos acceder mediante la barra de pestañas en la base de la aplicación.



16: Barra de pestañas

Añadir edificios

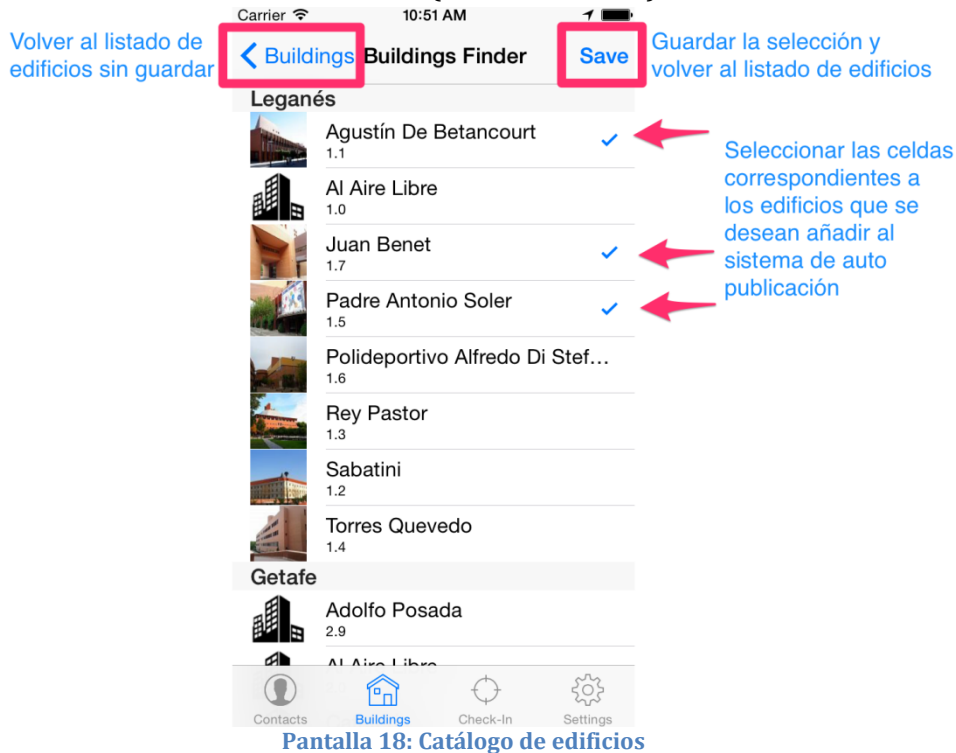
Desde la pantalla de edificios el usuario debe pulsar en la barra de navegación el botón + para acceder al catálogo de edificios de la UC3M.



Pantalla 17: Edificios añadidos

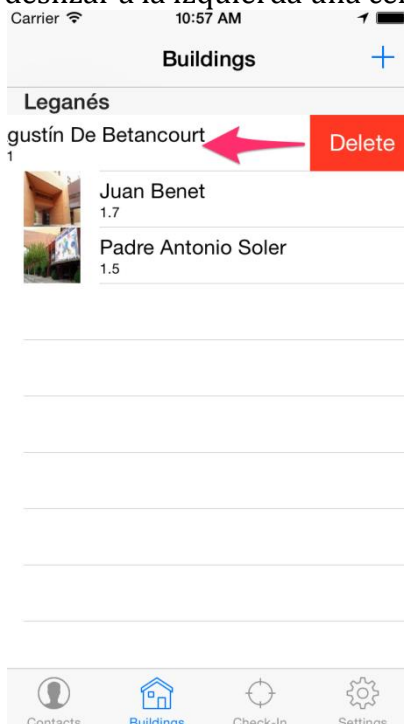
Tras la pulsación debe aparecer el catálogo para que el usuario seleccione los edificios en los que desee que se publique su llegada.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



Eliminar edificios

Para eliminar un edificio es necesario pulsar el botón “Delete” que se muestra al deslizar a la izquierda una celda

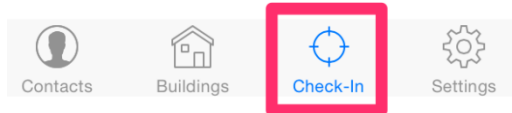


Checkin

Esta pantalla se usa para hacer checkins manuales para los lugares interiores de los edificios cercanos al usuario.

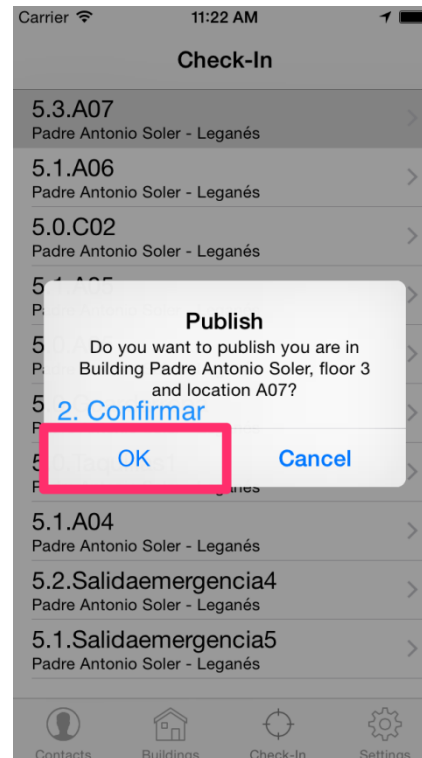
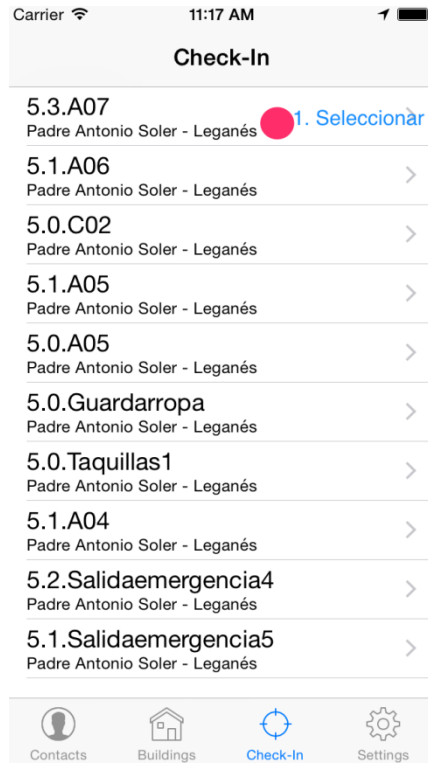
Para acceder a ella se usa la barra de pestañas en la base de la aplicación.

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)



20: Barra de pestañas

En esta pantalla se muestra un catálogo de estancias cercanas al usuario que pueden seleccionarse



Pantalla 21: Catálogo de estancias

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Anexo II: Plan de trabajo

Contenido

ANEXO II: PLAN DE TRABAJO.....	124
PLANIFICACIÓN	124
RECURSOS EMPLEADOS.....	125
<i>Recursos hardware.....</i>	<i>125</i>
<i>Recursos software.....</i>	<i>125</i>

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

Anexo II: Plan de trabajo

Planificación

En el diagrama de Gantt se puede ver la estimación de todas las fases del desarrollo. Al llevarse a cabo mediante una metodología iterativa con el desarrollo de cada historia de usuario se cubre un ciclo completo de diseño, desarrollo y pruebas para la historia de usuario que se entrega al finalizar la iteración. Al ser desarrollado por una persona las historias son resueltas de una en una hasta terminar el conjunto total de historias.

Esta planificación muestra un esfuerzo de 23 semanas. Al ser este el primer proyecto usando estas tecnologías resulta complicado hacer una estimación certera. La estrategia seguida para realizarla ha sido hacer una estimación de complejidad. De este modo se estima que el módulo de la gestión de contactos iba a ser el más complejo, el de las settings el más sencillo y en relación a estos, los demás asignando una serie de días en función de la complejidad estimada.

Al ser una única persona la que desarrolla el proyecto es difícil que la documentación se desarrolle en los tiempos que se ve en el Gantt pero se incluye aquí para mostrar una estimación de esfuerzo. En este caso si se tiene algo de experiencia en el desarrollo de esta documentación por lo que ha sido más sencillo de estimar.

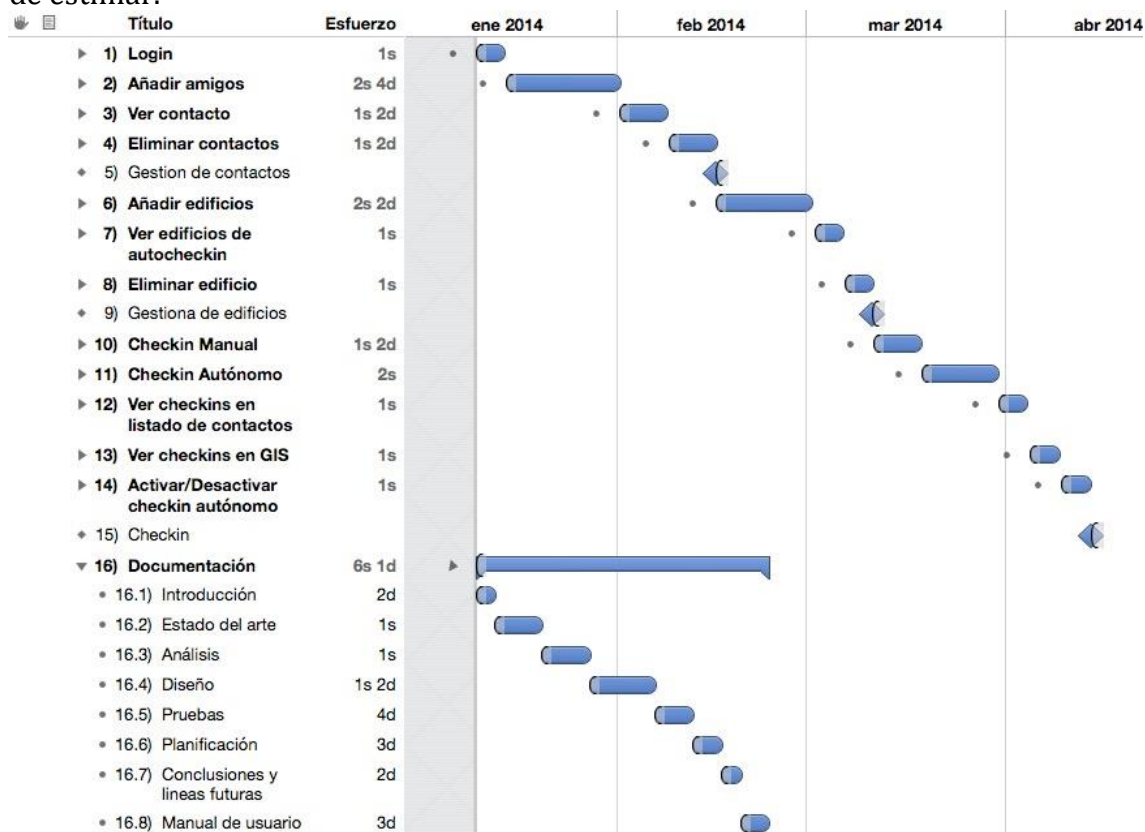


Ilustración 11: Plan de estimación de esfuerzo

Se han añadido una serie de hitos que representan el término de cada módulo de la aplicación. De esta manera cuando se termina un hito se ponen a disposición del

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

usuario todas las actividades definidas en cada historia de usuario que incluye el módulo terminado.

Gestión de contactos es el hito que da por terminado el módulo de la aplicación que permite añadir, eliminar y ver a los contactos del usuario. Para realizar estas actividades el usuario necesita estar registrado en el sistema, por lo que se añade el login en este grupo.

Gestión de edificios es el hito que da por terminado el módulo que permite al usuario añadir edificios para auto-checkin y eliminarlos

Checkin es el hito con el que se termina el desarrollo de la aplicación e incluye todas las actividades relacionadas con el checkin: Checkin autónomo en los edificios seleccionados por el usuario, checkin manual, la observación de las posiciones de los contactos en un mapa, la visualización de las posiciones en el listado de contactos y la opción de desactivar el checkin automático.

Recursos empleados

Los recursos empleados para este proyecto se dividen entre recursos software y recursos hardware.

En los recursos hardware se agrupan los medios físicos empleados directamente para realizar el proyecto y los recursos software agrupan las aplicaciones, herramientas software, recursos y servicios externos como la infraestructura donde se despliega la aplicación

Recursos hardware

Los recursos hardware empleados se limitan a un ordenador portátil y a un teléfono donde se prueba la aplicación. Se detallan a continuación.

Dispositivo	MacBook Pro
Procesador	2.8 GHz Intel Core i7
Memoria	16GB 1600MHz DDR3
Gráficos	Intel Iris 1536MB

Dispositivo	iPhone 5s
Procesador	1.3 GHz ARM v8
Memoria	1GB 1600MHz DDR3
Gráficos	PowerVR G6430

Recursos software

A continuación se muestra un listado con todo el software utilizado para la realización del proyecto. En él se encuentran las aplicaciones usadas para escribir la documentación, preparar la presentación, editar y mantener el código, hacer el seguimiento del proyecto buscar información y leerla.

Sistema operativo	OS X Yosemite
Procesador de textos	Microsoft Office Word 2011
Hoja de calculo	Microsoft Office Excel 2011
Presentación	Microsoft Office Power Point 2011
Entorno de programación	XCode 6.2
Entorno de programación	AppCode 3.1
Entorno de programación	Text Mate 2.0 Beta

Sistema de geolocalización de personas en interiores de grandes edificios
(Plataforma IOS)

Control de versiones	Git
Control de versiones	Bitbucket
Edición de imágenes	Gimp 2.8
Edición de imágenes	Skitch
Sistema de backup	Time Machine
Sistema de backup	Dropbox
Sistema de backup	Bitbucket
Visor de PDF	Vista Previa
Diagramas	Omni Plan, OmniGraffle
UML	Visual Paradigm
Navegador web	Safari
Servidor de aplicaciones Python 2.7	Google App Engine
Servicio de almacenamiento	Google Cloud Datastore

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)

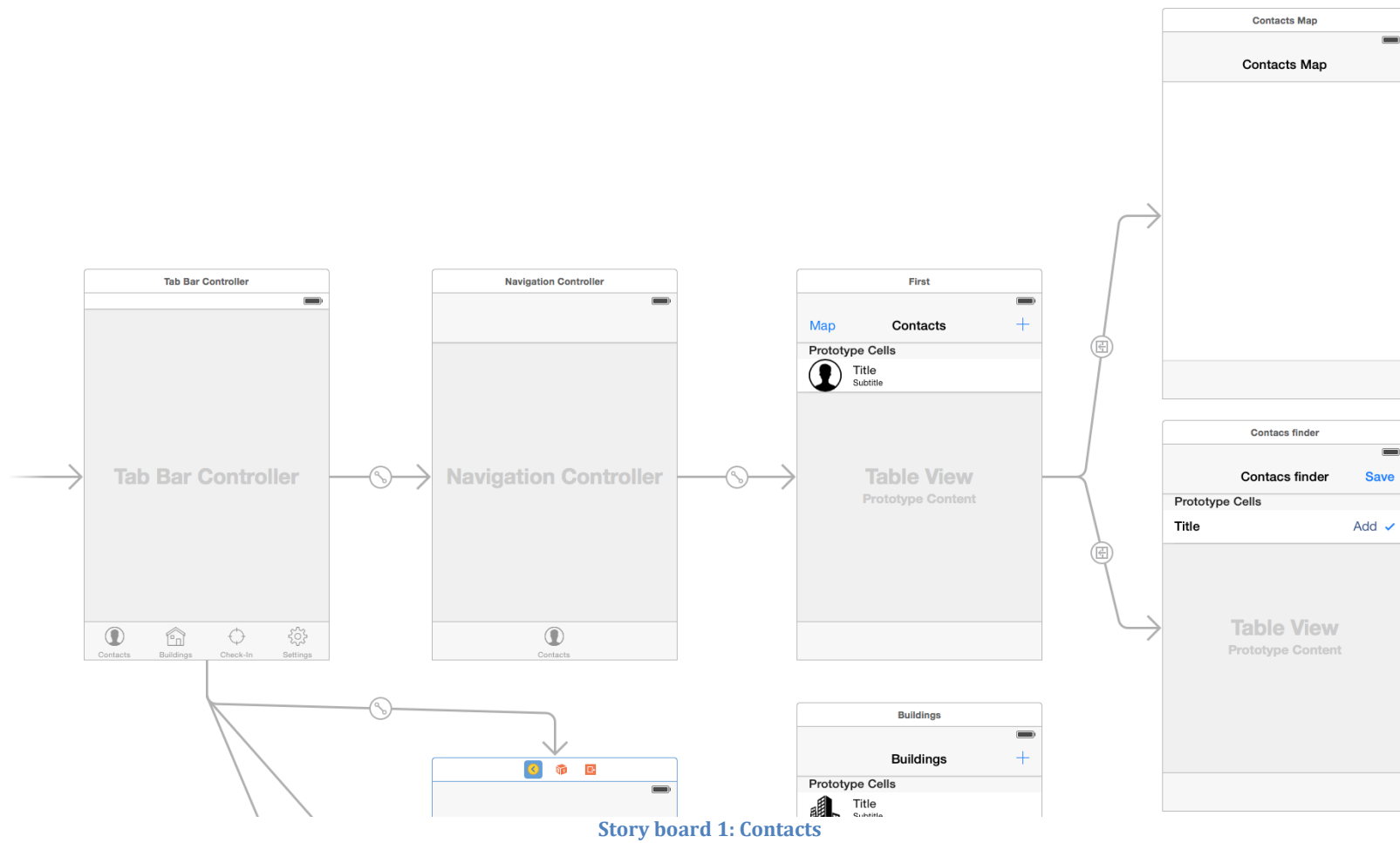
Anexo III: Story boards

Contenido

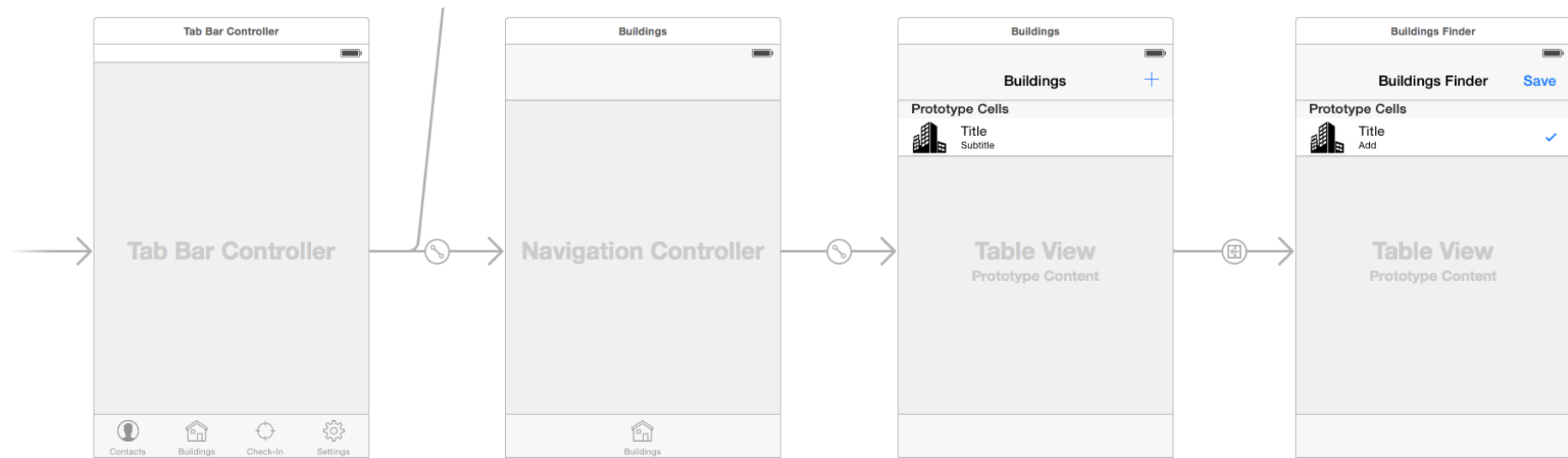
CONTACTS	129
BUILDINGS	130
CHECKIN	131
SETTINGS	132

Anexo III: Story board

Contacts

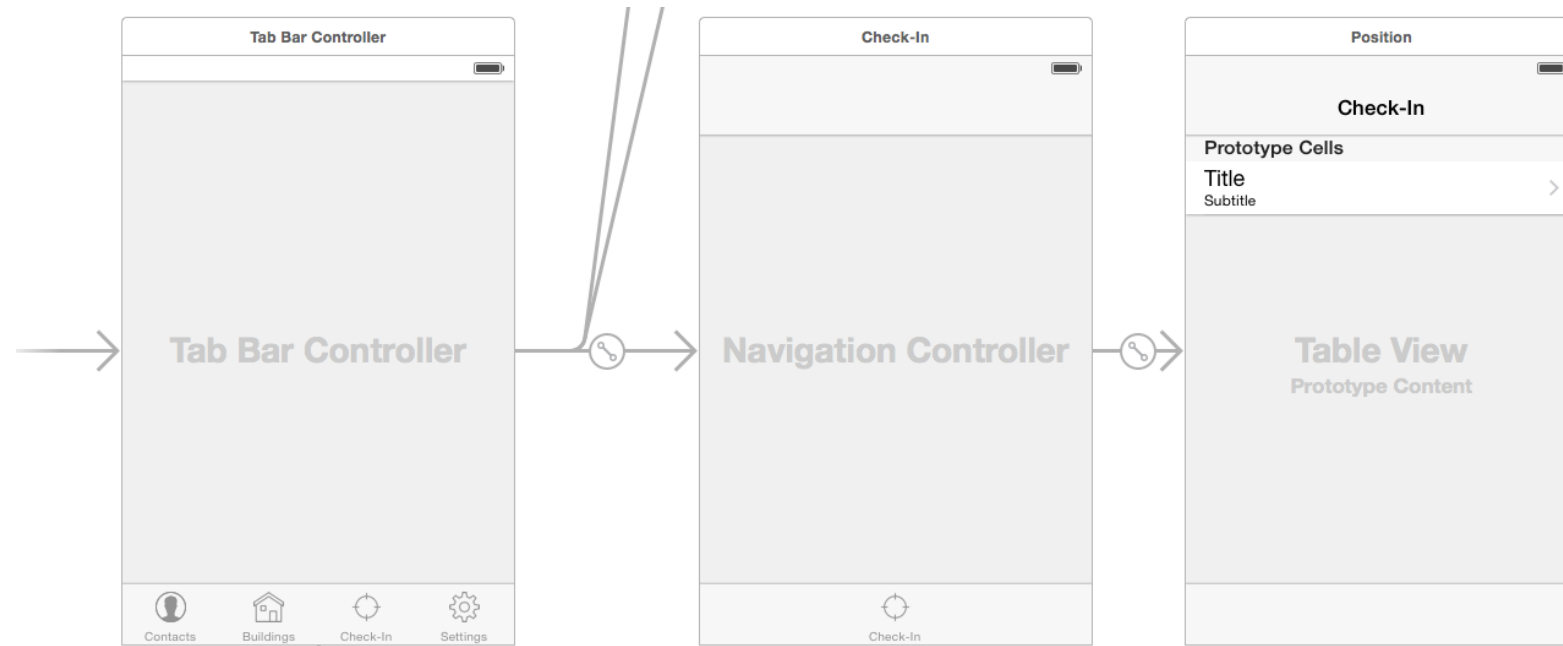


Buildings



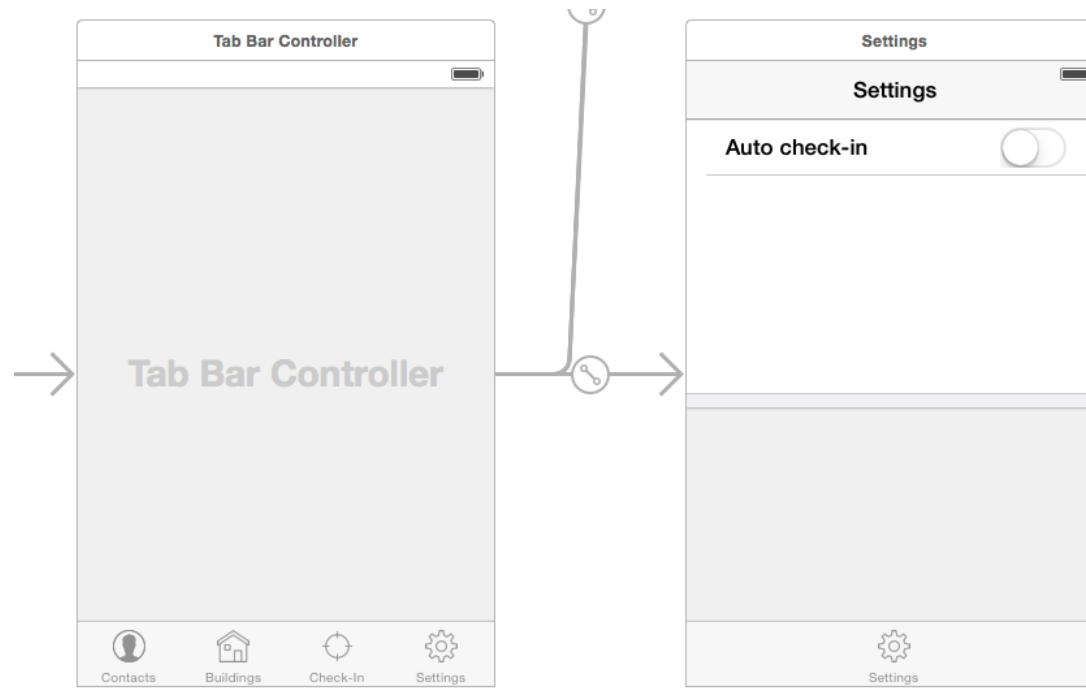
Storyboard 2: Buildings

Checkin



Storyboard 3: Checkin

Settings



Storyboard 4: Settings

Sistema de geolocalización de personas en interiores de grandes edificios (Plataforma IOS)